NASA Contractor Report 2846-2

Dynamic Loads Analysis System (DYLOFLEX) Summary

Volume II: Supplemental System Design Information

R. D. Miller, R. I. Kroll, and R. E. Clemmons

CONTRACT NAS1-13918
SEPTEMBER 1979

NASA

NASA Contractor Report 2846-2

Dynamic Loads Analysis System (DYLOFLEX) Summary

Volume II: Supplemental System Design Information

R. D. Miller, R. I. Kroll, and R. E. Clemmons
Boeing Commercial Airplane Company
Seattle, Washington

Prepared for
Langley Research Center
under Contract NAS1-13918

NASA
National Aeronautics
and Space Administration

Scientific and Technical
Information Branch

1979

CONTENTS

1.0 INTRODUCTION

DYLOFLEX consists of nine standalone programs that were developed in accordance with the requirements defined in reference 1. This document presents supplemental system design information pertaining to the DYLOFLEX program system. An engineering description of the DYLOFLEX system is found in volume I (ref. 2) of this document. Documentation for each DYLOFLEX program consists of two volumes: a user's guide and a supplemental system design and maintenance document. The user's guide for each program provides an engineering description, describes the input data required (cards and magnetic files), resources to be used (central processor seconds, print lines, etc.), and the job control cards needed to drive the program's execution. All user's guides are referenced in volume I of this document (ref. 2). The supplemental system design and maintenance documents for each program (see refs.) contain information concerning program structure and design; overlay purpose and description; input, output, and internal data base descriptions; and test cases. These volumes were written to aid those persons who will maintain and/or modify the programs in the future.

The information presented in this document explains the method of executing DYLOFLEX as a program system, the structure of magnetic files used to link the separate programs, and the various routines contained in the DYLOFLEX program library.

2.0 EXECUTION OF THE DYLOFLEX PROGRAM SYSTEM

When running an analysis through the DYLOFLEX system, the user may elect to run one module at a time, submitting a separate deck for each program after checking previous results. Once the system's operation is well understood, the user may then elect to run a series of modules driven by a single deck of job control cards. Figure 1 shows the control cards which might be used to execute both EOM(L217) and LOADS(L218) in a single run.

The magnetic files (tape or disk) provide the link tying the programs into a system. The files connecting the DYLOFLEX modules are displayed in figure 2. Note that default file names are listed. The names may be changed via card input data. Normally, during a program's execution the magnetic files will be in the form of disk files. Standard job control cards will be used to copy input data from tape to disk before executing the program, and output data to be saved will be copied from disk to tape after executing the program. This procedure provides a checkpoint/restart capability between each of the DYLOFLEX system's modules.

```
JOB CARD.
ACCOUNT CARD.
*
*     RETRIEVE THE PROGRAM L217(EOM) AND COPY IT TO
*     THE ABSOLUTE OVERLAY FILE L217.
*
REQUEST(MASTER,F=I,LB=KL,VSN=66XXXX)
REWIND(MASTER)
SKIPF(MASTER)
COPYBF(MASTER,L217)
RETURN(MASTER)
*
*     RETRIEVE THE INPUT MAGNETIC FILES REQUIRED FOR
*     L217(EOM). ALL HAD BEEN PREVIOUSLY WRITTEN ONTO
*     THE SAME TAPE , 66YYYY. THE AERODYNAMIC DATA
*     IS ASSUMED TO BE FROM L216(DUBFLX).
*
REQUEST(EOMDAT,F=I,LB=KL,VSN=66YYYY)
REWIND(EOMDAT)
COPYBF(EOMDAT,SSTIFF)
COPYBF(EOMDAT.GMASS)
COPYBF(EOMDAT,NGETP)
COPYBF(EOMDAT.NAETP)
COPYBF(EOMDAT,SATAP)
RETURN(EOMDAT)
*
*     EXECUTE L217(EOM)
*
RFL(120000)
L217.
*
*     SAVE THE STANDARD L217(EOM OUTPUT FILE
*
REQUEST(SL217,F=I,LB=KL)SAVE
REWIND(SL217,EOMTAP,EOMLOD)
COPYBF(EOMTAP,SL217)
COPYBF(EOMLOD,SL217)
RETURN(SL217)
RETURN(EOMTAP)
*
*     RETRIEVE THE PROGRAM L218(LOADS) AND COPY IT TO
*     THE ABSOLUTE OVERLAY FILE L218.
*
REQUEST(MASTER,F=I,LB=KL,VSN=66WWWW)
REWIND(MASTER)
SKIPF(MASTER)
COPYBF(MASTER,L218)
RETURN(MASTER)
*
*     RETRIEVE ADDITIONAL INPUT MAGNETIC FILRS REQUIRED BY
*     L218(LOADS).
*     L218(LOADS) WILL USE THE SAME INTERPOLATION DATA FILE,SATAP,
*     AND READ FROM EOMLOD.
*
REQUEST(LOADAT,F=I,LB=KL.VSN=66ZZZZ)
REWIND(LOADATA)
COPYBF(LOADAT,MASSTP)
RETURN(LOADAT)

*
*     EXECUTE L218(LOADS)
*
RFL(150000)
L218.
*
*     SAVE THE STANDARD L218(LOADS) OUTPUT FILES FROM A
*     V3MT RUN.
*
REQUEST(SL218,F=I,LB=KL)SAVE
REWIND(SL218)
REWIND(LTAP)
COPYBF(LTAP,SL218)
RETURN(SL218)
EXIT.
```

Figure 1.—Job Stacking in DYLOFLEX

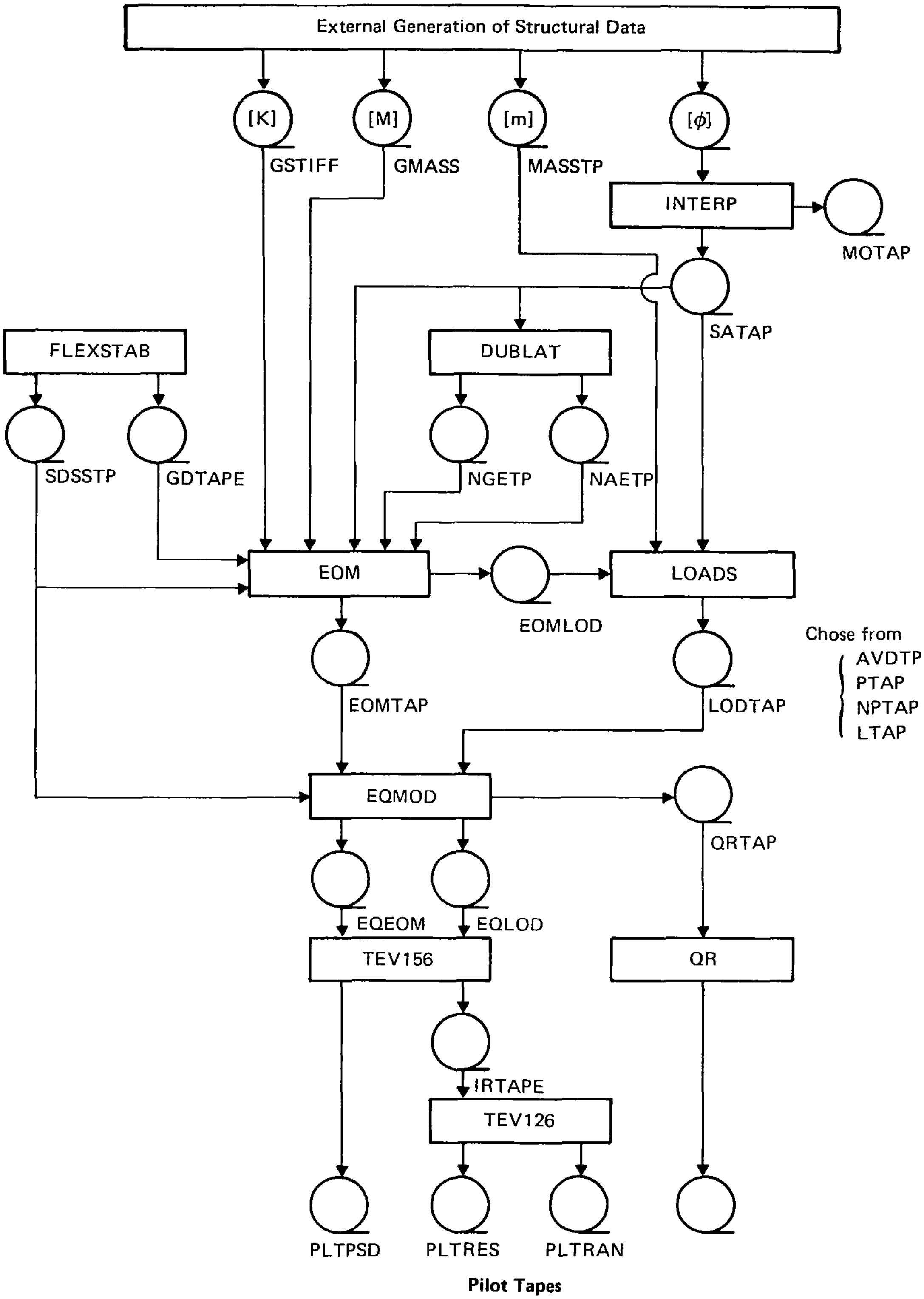

Figure 2.—Magnetic File Communication in DYLOFLEX

3.0 MAGNETIC FILE FORMAT

Almost all of the files connecting the DYLOFLEX programs are in the READTP/WRTETP format described in section 3.1. The exceptions are:

GDTAPE SDSSTP	The files generated by FLEXSTAB are described in reference 11, volume 2, appendix A.
NAETP	The aerodynamic data file generated by DUBFLX (L216) is also in the READTP/WRTETP format except for the records defining the matrices [F] and [D]-quasi-inverse.

Section 3.2 describes how the user may take any magnetic file readable by a FORTRAN program and convert it to the READTP/WRTETP format required by the DYLOFLEX programs. Section 3.3 describes how to reverse the process.

3.1 READTP/WRTETP FORMAT

The READTP/WRTETP format was designed to ease handling of matrix data. However, any variables in a FORTRAN program may be written/read onto/from a magnetic file as long as they are stored in adjacent core locations.

Each READTP/WRTETP matrix on a file is defined by two logical records. The first record, known as the "header" record, always contains 16 words.

Header Record Contents

Word	Variable	Description
1	NAME	Name assigned to the matrix
2	NROWS	Number of rows in the matrix
3	NCOLS	Number of columns in the matrix
4	0	Zero
5	ITYPE	Specifies the matrix storage type. All DYLOFLEX matrices have ITYPE = 0 indicating that all matrix elements are present in record two and they are stored row-wise.

Header Record Contents (Continued)

Word	Variable	Description
6	NELEM	Number of elements in the matrix
7-10	0	Zero
11	AUXID(1)	Words 1 through 6 from the auxiliary ID array input to READTP through the argument list (see sec. 6.1.2). AUXID allows up to six different variables to be carried along with the matrix on the magnetic file.
12	AUXID(2)	
13	AUXID(3)	
14	AUXID(4)	
15	AUXID(5)	
16	AUXID(6)	

The HEADER recorder is written onto the magnetic file with the FORTRAN unformatted write statement:-

WRITE (IFILE) (HEADER (I), I = 1,16)

The second record contains the matrix elements in row-wise order. The record is written with the FORTRAN unformatted write statement:

WRITE (IFILE) ((ELEM(I,J), J = 1,NCOLS), I = 1,NROWS)

3.2 CONVERTING FILES TO READTP/WRTETP FORMAT

Any digital data file which can be read by a FORTRAN program and which can be broken into sequential blocks may be represented by a READTP/WRTETP formatted file.

The user will have to write a small FORTRAN program which reads data from the existing file and stores the data in standard FORTRAN arrays of one or two dimensions (arrays of three or more dimensions will have to be broken into partitions of one or two dimensions). The program will then have to call the subroutine WRTETP to write the data onto the output file in the READTP/WRTETP format. The WRTETP subroutine may be obtained from the DYLOFLEX alternate subroutine library named DYLIB which is discussed in section 4.0.

Usage of WRTETP

The usage of WRTETP is discussed briefly in the following paragraphs. For a complete description of the routine, please see the comments at the beginning of the routine's source code.

```
DIMENSION AUXID(16)

DIMENSION MATRIX(NROWD,NCOLD)
```

Generate the element of MATRIX. Optionally store variables in AUXID(i): i=1,6. Optionally set NAME to some matrix identifier.

```
CALL WRTETP(MATRIX,NROWD,NAME,NROWS,NCOLS,AUXID,NFILES,
            NMATS,IFILE,IRR)
```

Input to WRTETP

MATRIX = The FORTRAN array containing matrix elements

NROWD = The dimensioned number of rows in the array MATRIX

NAME = Matrix identifier

NROWS
NCOLS = The actual number of rows and columns of MATRIX to be written onto the file

NROWS ≤ NROWD

NCOLS ≤ NCOLD

AUXID = Array of auxiliary identification data (only elements 1-6 will be retained on the file)

NFILES = Number of logical files to be skipped before writing the matrix on the file

NMATS = Number of matrices to be skipped before writing the matrix on the file

Note: Because each matrix is two logical records, 2*NMATS logical records will be skipped.

IFILE = Name of the file on which to write

Output from WRTETP

IRR = Error code

0 indicates no error detected

2 indicates NFILES or NMATS ≤ 0

3 indicates NROWD > 0

4 indicates NROWS $>$ NROWD

6 indicates NROWS*NCOLS ≤ 0

1000+I indicates an end-of-file was read after skipping (NFILES-I) files

3000+I indicates an end-of-file was read after skipping (2NMATS-I) logical records

3.3 CONVERTING FROM READTP/WRTETP FORMATTED FILES

Magnetic files in the READTP/WRTETP format may be reformatted by writing a small FORTRAN program. The program will have to call the subroutine READTP to read each matrix into core, and then the program may rewrite the information on another file with any method chosen by the authors of the program. The subroutine READTP may be obtained from the DYLOFLEX alternate subroutine library named DYLIB which is discussed in section 4.0.

Usage of READTP

The usage of READTP is discussed briefly in the following paragraphs. For a complete description of the routine, please see the comments at the beginning of the routine's source code.

```
DIMENSION AUXID(16)

DIMENSION MATRIX(NROWD,NCOLD)

Optionally set NAME = 0 to disable name check

CALL READTP (MATRIX,NROWD,NAME,NROWS,NCOLS,AUXID,NFILES,
             NMATS,IFILE,IRR)
```

Input to READTP

NROWD = The dimensioned number of rows in the FORTRAN array MATRIX to receive the matrix elements

NAME = The name against which the matrix name on file will be checked. If NAME = 0 no check is made

NFILES = Number of logical files to skip before reading

NMATS = Number of matrices to skip (after skipping files) before reading

IFILE = Name of the file from which to read

Output from READTP

MATRIX = The matrix elements

NAME = The matrix name on tape

NROWS
NCOLS = The actual number of rows and columns in the matrix from tape

$NROWS \leqslant NROWD$

$NCOLS \leqslant NCOLD$

AUXID = The auxiliary identification data. AUXID(7) - AUXID(12) will contain the six variables input to READTP through AUXID(1) - AUXID(6).

IRR = ERROR CODE

0 indicates no error detected

2 indicates NFILES or NMATS < 0

3 indicates NROWD < 0

4 indicates NROWS $>$ NROWD

5 indicates NAME did not match

6 indicates (NROWS*NCOLS) $\leqslant 0$ or > 10000

7 indicates an end-of-file was read when trying to read the matrix

1000+I indicates an end-of-information was read after skipping (NFILES-I) files

3000+I indicates an end-of-file was read after skipping (2*NMATS-I) logical records

4.0 DYLIB — THE DYLOFLEX ALTERNATE SUBROUTINE LIBRARY

All of the DYLOFLEX programs except the FLEXSTAB SD&SS module require the DYLOFLEX alternate subroutine library named DYLIB. All of the routines within DYLIB are described by comments in the PREFACE section of their source code. A copy of each PREFACE is given in Appendix A.

Table 1 provides a map showing which DYLIB routines are called by other DYLIB routines.

Table 1.—Map of Routines Called By DYLIB Routines

AINTG	BEAMO	ARCL	
		COMCUB	
		DATSRT	
		HERMINT	
		ZEROCOL	
	GTOLT	(entry point in AINTT)	
	LTOGT	(entry point in AINTT)	
	MOTAXO	DHRMINT	
		HERMINT	
		MAATTCH	ARCL
			REFPT
		ZEROCOL	
	MOTPTO	ZEROCOL	
	PLATEO	VIP	
		ZEROCOL	
	POLYO		
AINTL	BEAMO	ARCL	
		COMCUB	
		DATSRT	
		HERMINT	
		ZEROCOL	
	MOTAXO	DHRMINT	
		HERMINT	
		MAATTCH	ARCL
			REFPT
		ZEROCOL	
	MOTPTO	ZEROCOL	
	PLATEO	VIP	
		ZEROCOL	
	POLYO		

Table 1. — (Continued)

AINTT		
AMCON		
ARCL		
BEAMI	COMCUB	
	MADEFN	ARCL
		COMCUB
BEAMO	ARCL	
	COMCUB	
	DATSRT	
	HERMINT	
	ZEROCOL	
CBRT		
CDECOM	CINPRD	
	VIPDA	
CFBSUM	CINPRD	
	VIPDA	
CGLESM	CFBSUM	CINPRD
		VIPDA
	CDECOM	CINPRD
	VIPDA	
DINPRD	VIPA	
COMCUB		
DATSRT		
DECOM	AMCOM	
	VIPDA	
DELETR	LOCATR	
	XFER	
DELFIT	GET*	
	IRQL	
DHRMINT		
FBSUBM	VIPDA	
FETAD	FILESQ*	
	PUTFIT	
FETADD	FILESQ*	
	PUTFIT	
FETDEL	DELFIT	IRQL
FLUSH		

*Indicates a routine from the FORTRAN library
All others are DYLIB routines.

Table 1. — (Continued)

FNDKEY	IRDCRD	
FRTURN		
FSF	FSR	
FSR		
GETT		
GLESOM	DECOM	AMCOM
	FBSUBM	
	VIPDA	
HERMINT		
INITIR	DELETR	LOCATR
		XFER
	LOCATR	
	LOCF*	
IRDCRD		
IRQL	SHIFT	
ISCAN		
KNVRT	ISCAN	
	STRMOV	
KOMPAR		
KOMSTR		
LOCATR		
MAATTCH	ARCL	
	REFPT	
MADEFN	ARCL	
	COMCUB	
MOTAXI	COMCUB	
	MAATTCH	ARCL
		COMCUB
	MADEFN	ARCL
		COMCUB
MOTAXO	DHRMINT	
	HERMINT	
	MAATTCH	ARCL
		REFPT
	ZEROCOL	

*Indicates a routine from the FORTRAN library
All others are DYLIB routines.

Table 1. – (Continued)

MOTPTI			
MOTPTO	ZEROCOL		
NAMFIL	IRQL		
ORDER	VMIN		
PLATEA			
PLATEI	GLESOM	DECOM	AMCON
		FBSUBM	
		VIPDA	
	PLATEA		
	PLATET		
PLATEO	VIP		
	ZEROCOL		
PLATET			
POLYI			
POLYO			
PRGBEG	CLOCK*		
	DATE*		
PRGEND	CLOCK*		
	DATE*		
FLUSH			
PRINTM			
PRNTCM			
PUTFIT			
PUTT			
READTP	FSF		
	FSR		
	XPANDZ		
REFPT	CBRT		
REQFL			
	GET*		
RETURNF	FRTURN		
	IRQL		
	LOCF*		
RQL	SHIFT		
SEARCH			
STARTR	INITIR	DELETR	LOCATR
			XFER
		LOCATR	
	LOCF*		
	REQFL		

*Indicates a routine from the FORTRAN library
All others are DYLIB routines.

Table 1. – (Concluded)

STRMOV	
TBUL1	SEARCH
	TERP1
TERP1	
VIP	
VIPA	
VIPD	
VIPDA	
VIPDS	
VIPS	
VMIN	
WRTETP	FSF
	FSR
XFER	
XPANDZ	
ZEROCOL	

Boeing Commercial Airplane Company
P.O. Box 3707
Seattle, Washington 98124
May 1977

APPENDIX A
PREFACES FOR DYLIB SUBROUTINES

The following pages contain excerpts from the prefaces of each subroutine in DYLIB. The full preface is embedded in the code of each subroutine. A statement of the purpose of each subroutine and a brief description of how the subroutine works is given here. The subroutines are listed alphabetically.

AINTG

```
P U R P O S E
-------------

AINTG IS THE MODAL INTERPOLATION COVER ROUTINE FOR GENERATING DISPLAC-
EMENTS AND SLOPES AT AERODYNAMIC CONTROL POINTS SPECIFIED IN THE GLOB-
AL AXIS SYSTEM.

D E S C R I P T I O N
---------------------

AINTG PERFORMS ITS TASKS IN THE FOLLOWING STEPS
  (1) TRANSFORM OUTPUT POINTS TO LOCAL INTERPOLATION COORDINATE SYSTEM
      BY CALLING GTOLT ENTRY POINT OF AINTT.
  (2) SELECT INTERPOLATION OUTPUT ROUTINE ASSOCIATED WITH INTERPOLA-
      TION INFORMATION ARRAY.
  (3) GENERATE DISPLACEMENTS (OPTIONALLY, SLOPES) AT OUTPUT POINTS
      BY CALLING
      (A) PLATED  FOR SURFACE SPLINE METHOD
      (B) BEAMD   FOR BEAM SPLINE    METHOD
      (C) MOTAXD  FOR MOTION AXIS    METHOD
      (D) MOTPTD  FOR MOTION POINT   METHOD
      (E) POLYD   FOR POLYNOMIAL     METHOD
  (4) MODIFY DISPLACEMENTS (SLOPES) BY DIFFERENCE BETWEEN DIHEDRAL AT
      OUTPUT POINT AND ORIENTATION OF LOCAL SYSTEM W.R.T. GLOBAL
      SYSTEM.
  (5) TRANSFORM OUTPUT POINTS BACK TO GLOBAL SYSTEM BY CALLING LTOGT
      ENTRY POINT OF AINTT.
```

AINTL

P U R P O S E

AINTL IS THE MODAL INTERPOLATION COVER ROUTINE FOR GENERATING DISPLACEMENTS AND SLOPES AT AERODYNAMIC CONTROL POINTS SPECIFIED IN THE LOCAL AXIS SYSTEM.

D E S C R I P T I O N

AINTL PERFORMS ITS TASKS IN THE FOLLOWING STEPS
(1) SELECT INTERPOLATION OUTPUT ROUTINE ASSOCIATED WITH INTERPOLATION COEFFICIENT ARRAY (SA ARRAY)
(2) GENERATE DISPLACEMENTS (OPTIONALLY, SLOPES) AT OUTPUT POINTS BY CALLING
    (A) PLATEC   FOR SURFACE SPLINE METHOD
    (B) BEAMC   FOR BEAM SPLINE   METHOD
    (C) MOTAXC   FOR MOTION AXIS   METHOD
    (D) MOTPTC   FOR MOTION POINT   METHOD
    (E) POLYC   FOR POLYNOMIAL   METHOD

AINTT

P U R P O S E

AINTT IS THE COORDINATE TRANSFORMATION ROUTINE FOR THE AERODYNAMIC MODAL INTERPOLATION ROUTINE AINTG.

S U P E R I O R   R O U T I N E S
---------------   ---------------

AINTT IS CALLED BY AINTG

D E S C R I P T I O N

AINTT ENTRY POINTS ARE
(1) GTOLT - WHICH TRANSFORMS POINTS FROM GLOBAL TO LOCAL AXIS
(2) LTOGT - WHICH TRANSFORMS POINTS FROM LOCAL TO GLOBAL AXIS

AMCON

PURPOSE

PROVIDE SELECTIVE MACHINE AND MATHEMATICAL CONSTANTS WITH MAXIMUM ACCURACY.

DESCRIPTION

THE CONSTANTS ARE CODED AS ROUNDED OCTAL NUMBERS TO ACHIEVE MAXIMUM SIGNIFICANCE. THE AVAILABLE CONSTANTS ARE

SELECTOR J	DESCRIPTION	VALUE
1	LARGEST POSITIVE REAL	1.27*10**433
2	SMALLEST POSITIVE NORMALIZED REAL	3.13*10**-305
3	SMALLEST POSSIBLE REAL WHICH ADDS SIGNIFICANTLY TO 1.	7.11*10**-26
4	NUMBER OF BITS IN A WORD	60
5	NUMBER OF BITS IN MANTISSA OF A REAL	48
6	LARGEST POSITIVE NUMBER SUCH THAT ALL POSITIVE INTEGER VALUES LESS THAN OR EQUAL TO IT CAN CONVERT TO A FLOATING POINT NUMBER EXACTLY	2**59 - 1
7	NUMBER OF ALPHAMERIC CHARACTERS THAT CAN BE STORED IN ONE WORD	10
8,9	NOT USED	
10	BASE OF NATURAL LOGARITHMS	2.718...
11	PI	3.141...
12	LOG BASE 10 OF PI	0.4971...
13	LOG BASE E OF PI	1.144...
14	LOG BASE 10 OF E	0.4342...
15	EULERS CONSTANT	0.5772...
16	LOG BASE 2 OF E	1.442...
17	NUMBER OF RADIANS PER DEGREE	0.1745...
18	NUMBER OF DEGREES PER RADIAN	57.295...
19	1./E	0.367...

ARCL

P U R P O S E
- - - - - - -

CALCULATE THE ARC LENGTH ALONG THE CURVE X = C0+C1*Y+C2*Y**2+C3*Y**3
FROM (X0,Y0) TO (X,Y).

D E S C R I P T I O N
- - - - - - - - - - - -

THE ARC LENGTH IS GIVEN BY S = INTEGRAL& SQRT(1.+(DX/DY)**2*DY &.
IF C3=0, I.E. MOTION AXIS IS DEFINED BY QUADRATIC OR LOWER ORDER
POLYNOMIAL, A CLOSED FORM SOLUTION OF THE INTEGRAL IS EVALUATED.
IF THE MOTION AXIS IS A CUBIC, A FOUR POINT GAUSS-LEGENDRE QUADRATURE
IS USED TO NUMERICALLY EVALUATE THE INTEGRAL.

BEAMI

P U R P O S E

TO PERFORM AN INTERPOLATION IN TWO VARIABLES USING AS THE INTERPOLA-
TION FUNCTIONS, CUBIC SPLINES IN ARC LENGTH ALONG BEAMS (PLANAR
CURVES, X=F(Y)) WITH SECONDARY CUBIC SPLINES IN X GENERATED AT
REQUIRED Y VALUES USING THE VALUES DEFINED BY EACH BEAM.

D E S C R I P T I O N

BEAMI PERFORMS ITS TASKS IN THE FOLLOWING STEPS
  (1) INITIALIZE BASE POINTERS FOR COMPONENTS OF SA ARRAY
  (2) LOAD THE FOLLOWING INTO SA ARRAY
      A. FIRST 15 WORDS, VALUES FROM FORMAL ARGUMENT LIST
      B. BEAM POINTER AND EXTRAPOLATION CODES
      C. BEAM Y-COORDINATES
  (3) BEGIN LOOP ON NUMBER OF BEAMS
  (4) CALCULATE ARC LENGTH AND MAPPING POINT FOR ITH BEAM
  (5) BEGIN LOOP ON NUMBER OF MODES
  (6) CALCULATE SLOPES OF TZ IN ARC LENGTH AND LOAD INTO SA ARRAY
  (7) CALCULATE SLOPES OF RX IN ARC LENGTH AND LOAD INTO SA ARRAY
  (8) CALCULATE SLOPES OF RY IN ARC LENGTH AND LOAD INTO SA ARRAY
  (9) GO TO (5) FOR ANOTHER MODE
  (10) GO TO (3) FOR ANOTHER BEAM
  (11) LOAD 10HBEAMSPLINE INTO LAST WORD OF SA ARRAY
  (12) RETURN TO CALLING ROUTINE

BEAMO

P U R P O S E

TO CALCULATE MODAL DISPLACEMENTS AND OPTIONALLY SLOPES AT A SET OF OUTPUT AERODYNAMIC CONTROL POINTS, GIVEN AN INTERPOLATION INFORMATION ARRAY (SA ARRAY) GENERATED IN ROUTINE BEAMI FOR A BEAM-SPLINE SYSTEM

D E S C R I P T I O N

BEAMO PERFORMS ITS TASKS IN THE FOLLOWING STEPS
- (1) INITIALIZATION
  - (A) EXTRACT CONSTANTS FROM SA ARRAY
  - (B) CHECK VALUES OF PARAMETERS IF WITHIN LIMITS
  - (C) SET LEADING COLUMNS TO ZERO
  - (D) ESTABLISH POINTERS TO COMPONENTS OF SA ARRAY
  - (E) SET FIRST WORD OF EACH MODE SHAPE TO 9LUNDEFINED
- (2) BEGIN LOOP ON NUMBER OF POINTS
- (3) DETERMINE INTERSECTION POINTS OF A STREAMWISE CHORD DEFINED BY Y(I) AND EACH BEAM
- (4) COMPUTE THE ARC LENGTHS TO THE INTERCEPT POINTS
- (5) SORT X-COORDINATE VALUES MONOTONIC INCREASING AND ELIMINATE IDENTICAL POINTS
- (6) GENERATE SECONDARY SPLINE COEFFICIENTS AND DETERMINE DEFLECTIONS AND SLOPES FOR ALL OUTPUT POINTS ON THIS CHORD FOR EACH MODE
- (7) GO TO (2) FOR ANOTHER POINT
- (8) INITIALIZE TRAILING COLUMNS TO ZERO
- (9) RETURN TO CALLING ROUTINE

CBRT

P U R P O S E

CBRT FINDS THE CUBIC ROOT OF A REAL NUMBER XCUBED.

CDECOM

P U R P O S E

DECOMPOSE A SQUARE COMPLEX MATRIX INTO LOWER AND UPPER TRIANGULAR MATRICES WITH PARTIAL PIVOTING AND ROW EQUILIBRATION.

D E S C R I P T I O N

COMPLEX MATRIX A BECOMES DECOMPOSED INTO ITS LOWER AND UPPER TRIANGULAR COMPONENTS SUCH THAT A = L*U. SINGULARITY OF MATRIX A IS ALSO TESTED FOR AND A DIAGNOSTIC RETURNED. MATRIX A IS OVERLAYED BY L AND U. THE DIAGONAL ELEMENTS OF U ARE NOT STORED SINCE THEY EQUAL 1.

CFBSUM

P U R P O S E

SOLVE THE COMPLEX FORM OF A*X = B BY FORWARD AND BACKWARD SUBSTITUTIONS USING THE A = L*U DECOMPOSITION.

D E S C R I P T I O N

OBTAIN THE CROUT DECOMPOSITION OF COMPLEX MATRIX A AS LOWER AND UPPER TRIANGULAR MATRICES AND AN ARRAY OF INTEGERS WHOSE ELEMENT I IDENTIFIE PIVOTAL ROW I. SOLVE LOWER*Y = B AND UPPER*X = Y BY FORWARD AND BACKWARD SUBSTITUTIONS.

CGLESM

P U R P O S E

SOLVE THE COMPLEX FORM OF A*X = B FOR X, WHERE A IS AN N*N MATRIX AND B IS AN N*M MATRIX.

D E S C R I P T I O N

PERFORM THE CROUT DECOMPOSITION A = LU WITH ROW INTERCHANGES. IF A IS NOT SINGULAR, SOLVE THE TRIANGULAR SYSTEM BY FORWARD AND BACKWARD SUBSTITUTION. THE MATRICES ARE COMPLEX.

CINPRD

P U R P O S E

CALCULATE A = A - X*Y WHERE A IS A COMPLEX SCALAR AND X AND Y ARE COMPLEX VECTORS

D E S C R I P T I O N

THE USER SPECIFIES THE REAL AND IMAGINARY PARTS OF THE VARIABLES SEPARATELY. CINPRD COMPUTES THE REAL AND IMAGINARY PARTS SEPARATELY.

P U R P O S E

FIND THE SLOPES AT A GIVEN SET OF POINTS OF THE CUBIC SPLINE PASSING THROUGH THE POINTS SATISFYING EITHER USER SPECIFIED END CONDITIONS OR INTERNALLY COMPUTED END CONDITIONS.

D E S C R I P T I O N

```
THE APPROACH IS TO GENERATE A SET OF N SIMULTANEOUS EQUATIONS AND
SOLVE.    (N-2) OF THE EQUATIONS RESULT FROM THE CONTINUITY CONDITIONS
AT EACH INDEPENDENT VARIABLE POINT. FOR DATA POINT J, THE EQUATION IS
 M(J)/H(J) + 2.*(1./H(J) + 1./H(J+1))*M(J) + M(J+1)/H(J+1) =
 3.*(Y(J) - Y(J-1))/H(J)**2
WHERE M(J) = SLOPE AT X(J)
      H(J) = X(J) - X(J-1)
THE OTHER 2 EQUATIONS RESULT FROM THE END CONDITIONS. FOR DATA POINT 1
1.    M(1) = SPECIFIED FIRST DERIVATIVE, OR
2.    2.*M(1) + M(2) = 3.*(Y(2) - Y(1))/(X(2) - X(1)) - (X(2) - X(1))/
        2.*SPECIFIED SECOND DERIVATIVE, OR
3.    M(1) = A(1)+A(2)*M(2) WHERE M(1) IS INTERPOLATED, OR
4.    2.*M(1) + MU*M(2) = C WHERE THE SECOND DERIVATIVE IS INTERPOLATE
            AND MU = 4.*(1.+2.*A(2))/(4.+2.*A(2))
                 C = -2.*A(1)*(X(2)-X(1))/(4.+2.*A(2)) + 6.X(2.+2.*
                     A(2))/(4.+2.*A(2)))*(Y(2)-Y(1))/(X(2)-X(1))
FOR DATA POINT N,
1.    M(N) = SPECIFIED FIRST DERIVATIVE, OR
2.    M(N-1) + 2.*M(N) = 3.*(Y(N)-Y(N-1))/(X(N)-X(N-1)) +
        (X(N)-X(N-1))/2.*SPECIFIED SECOND DERIVATIVE, OR
3.    M(N) = B(1) + B(2)*M(N-1) WHERE M(N) IS INTERPOLATED, OR
4.    4.*(1.+2.*B(2))/(4.+2.*B(2))*M(N-1) + 2.*M(N) = C  WHERE
        THE SECOND DERIVATIVE IS INTERPOLATED AND
                 C = 2.*B(1)*(X(N)-X(N-1))/(4.+2.*B(2)) +
                   6.*(1.+B(2))/(2.+B(2))*(Y(N)-Y(N-1))/(X(N)-
                     X(N-1))
IF THE END CONDITIONS ARE NOT EASILY ESTIMATED, SET THE FOLLOWING
      KA = KB = -2
      A(1) = B(1) = 0.
      A(2) = B(2) = .5  FOR INCREASING CURVATURE, OR
      A(2) = B(2) = -.5  FOR DECREASING CURVATURE
```

THE RESULTING MATRIX IS TRIDIAGONAL AND DIAGONALLY DOMINANT. THE EQUATIONS ARE SOLVED BY GAUSSIAN ELIMINATION WITH NO PIVOTING, MAKING FULL USE OF THE SPARSENESS OF THE MATRIX.

DATSRT

PURPOSE

DATSRT REORDERS AN ARRAY OF NUMBERS ASCENDING AND ELIMINATES NUMBERS FROM THE ARRAY ACCORDING TO THE FOLLOWING CONDITIONS (1) VALUE EQUAL SLNOT USED AND (2) VALUE EQUAL PREVIOUS VALUE (I.E. IF TWO VALUES ARE EQUAL, ELIMINATE ONE)

SUPERIOR ROUTINES

DATSRT IS CALLED BY BEAMO

DESCRIPTION

DATSRT PERFORMS ITS FUNCTIONS IN THE FOLLOWING STEPS
(1) INITIALIZE POINTERS TO ARRAY OF NUMBERS AND INDEX MAP
(2) ELIMINATE NOT USED VALUES AND COMPRESS ARRAY OF NUMBERS
(3) REORDER VALUES ASCENDING AND RETAIN INDEX MAP
(4) ELIMINATE A VALUE WHEN TWO VALUES ARE WITHIN AN EPSILON
(5) RETURN TO CALLING ROUTINE

DECOM

PURPOSE

DECOMPOSE A SQUARE MATRIX INTO LOWER AND UPPER TRIANGULAR MATRICES WITH PARTIAL PIVOTING AND ROW EQUILIBRATION.

DESCRIPTION

MATRIX A BECOMES DECOMPOSED INTO ITS LOWER AND UPPER TRIANGULAR COMPONENTS SUCH THAT A = L*U. SINGULARITY OF MATRIX A IS ALSO TESTED FOR AND A DIAGNOSTIC RETURNED. MATRIX A IS OVERLAYED BY L AND U. THE DIAGONAL ELEMENTS OF L ARE NOT STORED SINCE THEY ARE EQUAL TO 1.

DELETR

PURPOSE

DELETR IS CALLED TO ELIMINATE THE ARRAY NAMEA FROM THE LIBRARY AND STORAGE.

DESCRIPTION

ROUTINE DELETR WILL ACCOMPLISH ITS TASK IN THE FOLLOWING STEPS...

A) INITIALIZE THE ERROR CODE.

B) HUNT FOR NAMEA IN THE VARDIM (VARIABLE DIMENSION) CATALOG. IF NAME IS NOT FOUND THE ERROR CODE IS SET TO -2 AND WILL JUMP TO (D). IF NAMEA IS THE LAST ENTRY, THEN NAMEA WILL BE ERASED AND A JUMP TO (D) WILL TAKE PLACE.

C) OTHERWISE COMPRESS THE CORE STORAGE AND VARDIM CATALOG.

D) RETURN TO CALLING PROGRAM.

DELFIT

P U R P O S E

DELFIT IS CALLED TO DELETE A FILE NAME FROM THE SYSTEMS RA+2 LIST OF FILE NAMES. BEFORE DELETING THE NAME IT MAKES SURE THAT THE BUFFER OF AN OUTPUT FILE HAS BEEN EMPTIED.

S U P E R I O R   R O U T I N E S

FETDEL   CYLIB   CLOSE A FILE AND DELETE THE FILES ENTRY IN THE RA+2 LIST.

D E S C R I P T I O N

DELFIT PERFORMS ITS TASKS IN THE FOLLOWING STEPS...

```
  (1) INITIALIZE THE ERROR CODE.
  (2) FIND THE FILE NAME IN THE RA+2 LIST.
      IF NOT PRESENT, SET THE ERROR CODE TO -1 AND JUMP TO (9).
  (3) CHECK THE FILES FIT TO SEE IF THE LAST OPERATION WAS A WRITE.
      IF NOT, JUMP TO (7).
      IF YES, BUT THE BUFFER IS EMPTY JUMP TO (7).
  (4) CHECK THE FIT AGAIN TO SEE IF THE FILE IS OF TYPE WORD ADDRESS.-
      ABLE.  IF NOT JUMP TO (6).
  (5) READ THE FIRST WORD ON THE WORD ADDRESS ABLE (RANDOM) FILE.
      THEIS WILL FORCE THE BUFFER TO BE EMPTIED.
      THIS  WILL FORCE THE BUFFER TO BE EMPTIED.
  (6) REWIND THE FILE.
      JUMP TO (7).
  (7) DELETE THE ENTRY IN THE RA+2 LIST.
  (8) COMPRESS THE RA+2 TO REMOVE IMBEDDED ZERO.
  (9) RETURN TO THE CALLING PROGRAM.
```

DHRMINT

P U R P O S E

GIVEN F(X1), F(X2), DF(X1)/DX, AND DF(X2)/DX, CALCULATE DF(X)/DX, X2&X&X1, FOR N MODES, WHERE HERMITE INTERPOLATION IS USED TO APPROXIMATE F(X).

S U P E R I O R   R O U T I N E S

MUTAXC - CALCULATE INTERPOLATED VALUES AT A SET OF OUTPUT POINTS

D E S C R I P T I O N

THE HERMITE INTERPOLATION FORM IS

```
F(X) = C1& C2(C3*DF(X1)/DX + C4*DF(X2)/DX) + C5(C6*F(X1) + C7*F(X2)) &

      C7 = 2(X2-X) - (X2-X1)
      C6 = -(X2-X)*(X2-X1)
      C5 = (X-X1)**2
      C4 = 2(X-X1) - (X2-X1)
      C3 = (X-X1)*(X2-X1)
      C2 = (X2-X)**2
      C1 = 1./(X2-X1)**3
```

THE DERIVATIVE OF THE HERMITE FORM WITH RESPECT TO X IS,

```
DF(X)/DX = C1&DC2(C3*DF(X1)/DX + C4*F(X1)) + DC5(C6*DF(X2)/DX +
               C7F(X2)) + C2(DEL*DF(X1)/DX + 2*F(X1))
             + C5(DEL*DF(X2)/DX - 2*F(X2)) &

          DC2 = -2.(X2-X)
          DC5 =  2.(X-X1)
          DEL =  X2-X1
```

FBSUBM

P U R P O S E

SOLVE THE EQUATION A*X = B BY FORWARD AND BACKWARD SUBSTITUTIONS USING THE A = L*U DECOMPOSITION.

D E S C R I P T I O N

OBTAIN THE CROUT DECOMPOSITION OF MATRIX A AS LOWER AND UPPER TRIANGULAR MATRICES AND AN ARRAY OF INTEGERS WHOSE ELEMENT I IDENTIFIES PIVOTAL ROW I. SOLVE LOWER*Y = B AND UPPER*X = Y BY FORWARD AND BACKWARD SUBSTITUTIONS.

FETAD

P U R P O S E

FETAD PERMITS A FORTRAN PROGRAM TO DEFINE A FILE DURING EXECUTION. THUS, IF A FILE IS USED ONLY DURING ONE PHASE OF A PROGRAM THE FIELD LENGTH ASSOCIATED WITH ITS BUFFER NEED NOT BE DEDICATED TO THAT FILE THROUGHOUT EXECUTION IT COULD BE USED AS A BUFFER FOR ANOTHER FILE OR ANY OTHER PURPOSE. SUBROUTINE FETDEL COMPLIMENTS THIS ROUTINE AND IS USED TO CLOSE AND DELETE A FILE.

D E S C R I P T I O N

IF IFILE IS AN INTEGER IN THE RANGE OF 1 TO 99, ITS NUMERIC VALUE, NN. IS USED TO GENERATE THE DISPLAY CODE FILE NAME TAPENN. OTHERWISE, IFILE IS USED DIRECTLY.

PUTFIT IS USED TO ADD THE FILE NAME TO THE RA+2 LIST WITH THE POINTER SPECIFYING FET(1) AS THE START OF THE FIT. FILESQ IS CALLED TO DESCRIBE THE FILE TO THE RECORD MANAGER WITH BUFFER(1) AS THE START OF THE FILES BUFFER.

FETADD

P U R P O S E

FETADD PERMITS A FORTRAN PROGRAM TO DEFINE A FILE DURING EXECUTION. THUS, IF A FILE IS USED ONLY DURING ONE PHASE OF A PROGRAM THE FIELD LENGTH ASSOCIATED WITH ITS BUFFER NEED NOT BE DEDICATED TO THAT FILE THROUGHOUT EXECUTION IT COULD BE USED AS A BUFFER FOR ANOTHER FILE OR ANY OTHER PURPOSE. SUBROUTINE FETDEL COMPLIMENTS THIS ROUTINE AND IS USED TO CLOSE AND DELETE A FILE.

D E S C R I P T I O N

IF IFILE IS AN INTEGER IN THE RANGE OF 1 TO 99, ITS NUMERIC VALUE, NN. IS USED TO GENERATE THE DISPLAY CODE FILE NAME TAPENN. OTHERWISE, IFILE IS USED DIRECTLY.

PUTFIT IS USED TO ADD THE FILE NAME TO THE RA+2 LIST WITH THE POINTER

SPECIFYING ARRAY(1) AS THE START OF THE FIT. FILESQ IS CALLED TO DESCRIBE THE FILE TO THE RECORD MANAGER WITH ARRAY(36) AS THE START OF THE FILES BUFFER.

FETDEL

P U R P O S E

FETDEL PERMITS A FORTRAN PROGRAM TO CLOSE AND DELETE A FILE DURING EXECUTION. IT ALLOWS DYNAMIC MANAGEMENT OF FILE NAMES AND BUFFERS. THUS, IF A FILE IS USED ONLY DURING ONE PHASE OF A PROGRAM THE FIELD LENGTH ASSOCIATED WITH ITS BUFFER NEED NOT BE DEDICATED TO THAT FILE THROUGHOUT EXECUTION IT COULD BE USED AS A BUFFER FOR ANOTHER FILE OR ANY OTHER PURPOSE. SUBROUTINES FETAD AND FETADD COMPLIMENT THIS ROUTINE AND ARE USED DEFINE A NEW FILE.

D E S C R I P T I O N

IF IFILE IS AN INTEGER IN THE RANGE OF 1 TO 99, ITS NUMERIC VALUE, NN. IS USED TO GENERATE THE DISPLAY CODE FILE NAME TAPENN. OTHERWISE, IFILE IS USED DIRECTLY.

PUTFIT IS USED TO ADD THE FILE NAME TO THE RA+2 LIST WITH THE POINTER SPECIFYING FET(1) AS THE START OF THE FIT. FILESQ IS CALLED TO DESCRIBE THE FILE TO THE RECORD MANAGER WITH BUFFER(1) AS THE START OF THE FILES BUFFER.

FNDKEY

P U R P O S E

FNDKEY READS AND PRINTS CARD INPUT IMAGES UNTIL IT FINDS ONE WITH A GIVEN KEY IN THE CARDS FIRST CHARACTERS.

D E S C R I P T I O N

FNDKEY PERFORMS ITS TASK IN THE FOLLOWING STEPS.
(1) INITIALIZE THE ERROR FLAG.
(2) CALL IRDCRD TO READ AN INPUT CARD, CHECK FOR END-OF-FILE, PRINT THE CARD IMAGE, AND EXTRACT THE KEYWORD.
JUMP TO (4) IF A END-OF-FILE WAS READ
(3) COMPARE THE ACTUAL KEYWORD (IWORD) AGAINST THE ONE DESIRED(KEY)
JUMP TO (4) IF IWORD .EQ. KEY.

REPEAT STEPS 2-3 IF .NE. KEY.

(4) RETURN TO CALLING PROGRAM.

FSF

P U R P O S E

FSF FORWARD SPACES PAST LOGICAL FILES ON SEQUENTIAL BINARY FILES.

NOTE ... THIS IS NOT THE BCS STANDARD VERSION DESCRIBED IN REF. 2.
THIS VERSION WAS WRITTEN FOR DYLOFLX (REF. 1).

D E S C R I P T I O N

FOR EACH LOGICAL FILE TO BE SKIPPED ON THE MAGNETIC FILE, FSF CALLS FSR TO SKIP PAST N0000 LOGICAL RECORDS OF AN END-OF-FILE. THE FSR ERROR CODE WILL BE GREATER THAN ZERO IF AN END-OF-FILE WAS READ. THE ERROR CODE WILL BE N0000 IF AN EMPTY FILE WAS FOUND.

ASSUMPTIONS ...
(1) NO LOGICAL FILE WILL CONTAIN MORE THAN N0000 LOGICAL RECORDS.
(2) SIX CONSECUTIVE EMPTY FILES ARE TAKEN TO MEAN END-OF-INFORMATION

LIMITATIONS ...
FSF MAY NOT PROCESS CORRECTLY MAGNETIC FILES WITH
(1) MORE THAN N0000 LOGICAL RECORDS IN A LOGICAL FILE.
(2) SIX OR MORE CONSECUTIVE EMPTY LOGICAL FILES.

FSR

P U R P O S E

FSR FORWARD SPACES PAST LOGICAL RECORDS ON SEQUENTIAL BINARY FILES.

D E S C R I P T I O N

FSR READS SHORT LIST (1 WORD) FORTRAN BINARY RECORDS UNTIL EITHER THE CORRECT NUMBER OF RECORDS HAVE BEEN SKIPPED OR AN END-OF-FILE IS READ. THE LATTER CASE RESULTS IN THE ERROR CODE BEING SET TO A NON-ZERO VALUE. BACK SPACING IS NOT ALLOWED. END-OF-FILES WILL NOT BE SKIPPED.

GLESOM

P U R P O S E

SOLVE A*X = B FOR X, WHERE A IS AN N*N MATRIX AND B IS AN N*M MATRIX.

D E S C R I P T I O N

PERFORM THE CROUT DECOMPOSITION A = LU WITH ROW INTERCHANGES. IF A IS NOT SINGULAR, THEN SOLVE THE TRIANGULAR SYSTEM BY FORWARD AND BACKWARD SUBSTITUTION.

HERMINT

P U R P O S E

GIVEN F(X1), F(X2), DF(X1)/DX, AND DF(X2)/DX FOR N MODES PERFORM HERMITE INTERPOLATION FOR F(X), X1&X&X2.

S U P E R I O R   R O U T I N E S

MOTAXO - CALCULATE INTERPOLATED VALUES AT A SET OF OUTPUT POINTS

D E S C R I P T I O N

THE HERMITE INTERPOLATION FORM IS

```
F(X) = C1& C2(C3*DF(X1)/DX + C4*DF(X2)/DX) + C5(C6*F(X1) + C7*F(X2)) &

      C1 = 1./(X2-X1)**3
      C2 = (X2-X)**2
      C3 = (X-X1)*(X2-X1)
      C4 = 2(X-X1) - (X2-X1)
      C5 = (X-X1)**2
      C6 = -(X2-X)*(X2-X1)
      C7 = 2(X2-X) - (X2-X1)
```

INITIR

P U R P O S E

INITIR IS CALLED TO INITIALIZE (DEFINE) A NEW ARRAY.

D E S C R I P T I O N

INITIR ALLOCATES THE ARRAY STORAGE, ZEROS THE AREA, AND MAKES AN ENTRY IN THE ARRAY CATALOGUE. IF THE ARRAY BEING DEFINED ALREADY EXISTS, THE ELEMENTS ARE SIMPLY SET TO ZERO IF THE DIMENSIONS ARE TO REMAIN UNCHANGED. IF THE ARRAY SIZE IS TO BE CHANGED THE OLD ARRAY IS DELETED AND A NEW ONE DEFINED.

NOTE... THE ROUTINE INITIR MUST BE CALLED FOR A ARRAY BEFORE ANY OTHER VARDIM ROUTINE CAN REFER TO THE ARRAY.

IRDCRD

P U R P O S E

IRDCRD READS A CARD IMAGE FROM THE FILE INFIL, PRINTS IT ON IUTFIL, AND EXTRACTS THE KEYWORD FROM ITS FIRST CHARACTERS.

D E S C R I P T I O N

```
IRDCRD PERFORMS ITS TASK IN THE FOLLOWING STEPS ...
 (1) INITIALIZE THE ERROR CODE.
 (2) READ A CARD IMAGE FROM THE FILE INFIL.
     IF AN END-OF-FILE IS READ THE ERROR CODE IS SET EQUAL TO 1,
                              AND 10H----EOF--- IS PLACED IN ICARD(1).
 (3) PRINT ICARD ON IUTFIL.
     JUMP TO STEP (5) IF IPR=1.
 (4) MASK THE KEYWORD FROM ICARD(1) INTO KEY WITH MASK.
     IF KEY =  L$TITL JUMP TO STEP (2).
 (5) RETURN TO THE CALLING PROGRAM.
```

KNVRT

P U R P O S E

SCAN A STRING OF CHARACTERS, EXTRACT NUMERIC STRINGS, CONVERT THEM TO BINARY NUMBERS AND STORE THEM IN AN OUTPUT ARRAY.

D E S C R I P T I O N

SCAN A STRING OF CHARACTERS AND IDENTIFY THE BEGINNING AND ENDING (A BLANK) OF A FIELD CONTAINING NUMERIC DATA. IDENTIFY THE NUMBER IN THE FIELD AS INTEGER OR REAL AND CONVERT IT TO THE APPROPRIATE FORMAT, VIZ I20, F20.0 OR E30.0. WHEN THE DESIRED NUMBER OF VALUES HAS BEEN CONVERTED OR THE ENTIRE STRING HAS BEEN SCANNED, CONTROL RETURNS TO THE CALLING PROGRAM.

LOCATR

P U R P O S E

LOCATR IS CALLED TO DETERMINE AN ARRAYS SIZE, TYPE, AND LOCATION.

D E S C R I P T I O N

ROUTINE LOCATR IS CALLED TO DETERMINE A SIZE, TYPE, AND LOCATION OF AN ARRAY IN VARDIM STORAGE. ROUTINE LOCATR SHOULD BE CALLED JUST PRIOR TO HANDLING THE ARRAY, BECAUSE ITS LOCATION CHANGES AS ARRAYS ARE DELETED AND ADDED TO THE VARIABLE DIMENSION STORAGE SCHEME.

MAATTCH

P U R P O S E

GIVEN THE DEFINITION OF A MOTION AXIS DETERMINE THE REFERENCE POINT, (XR,YR), ON THE MOTION AXIS AND SUNDRY OTHER INFORMATION ASSOCIATED WITH AN ARBITRARY POINT (X,Y).

S U P E R I O R   R O U T I N E S

MOTAXI - GENERATE AN INTERPOLATION INFORMATION ARRAY FOR A MOTION AXIS SYSTEM
MOTAXO - CALCULATE OUTPUT FOR A MOTION AXIS SYSTEM AT A SET OF POINTS

D E S C R I P T I O N

THE MOTION AXIS IS DEFINED BY A SERIES OF CUBICS BETWEEN MOTION AXIS DEFINITION POINTS. THE REFERENCE LINES THROUGH EACH OF THE MOTION AXIS DEFINITION POINTS DIVIDE THE XY PLANE INTO REGIONS WHICH ARE ASSOCIATED WITH EACH OF THE MOTION AXIS SEGMENTS. GIVEN AN ARBITRARY POINT, (X,Y), THE ASSOCIATED MOTION AXIS SEGMENT IS DETERMINED, AND THE CUBIC COEFFICIENTS ACCESSED, (OR CALCULATED FOR AN EXTRAPOLATION CONDITION). IN THE CASE OF A POINT WHICH IS NOT WITHIN THE REGIONS ASSOCIATED WITH A MOTION AXIS SEGMENT, I.E. INBOARD OR OUTBOARD, A LINEAR EXTRAPOLATION USING THE EXTREME POINT AND SLOPE OF THE MOTION AXIS IS USED TO DEFINE THE COEFFICIENTS. IF THE REFERENCE LINES ON EITHER END OF THE ASSOCIATED SEGMENT ARE PARALLEL, A LINE OF THE SAME SLOPE THROUGH (X,Y) MAPS (X,Y) TO THE SEGMENT. OTHERWISE A LINE THROUGH THE POINT OF INTERSECTION OF THE REFERENCE LINES, (XM,YM), AND (X,Y) MAPS (X,Y) TO THE SEGMENT.

IN ADDITION TO THE REFERENCE POINT, THE ARC LENGTH ALONG THE MOTION AXIS, THE DISTANCE FROM (X,Y) TO REFERENCE POINT, AND THE ANGULAR ORIENTATION OF THE REFERENCE LINE THROUGH (X,Y) ARE CALULATED. ALSO THE RATIO OF DISTANCE FROM (X,Y) TO (XR,YR) AND THE DISTANCE FROM (XM,YM) TO (X,Y), AND THE SLOPE OF THE MOTION AXIS AT (XR,YR) ARE CALCULATED.

MADEFN

P U R P O S E

GIVEN A SET OF MOTION AXIS DEFINITION POINTS AND THE SLOPE OF A REFERENCE LINE THROUGH EACH POINT, GENERATE A SPLINE OF CUBICS DESCRIBING THE MOTION AXIS, CALCULATE THE ARC LENGTH ALONG THIS CURVE TO EACH DEFINITION POINT, AND CALCULATE THE MAPPING POINT ASSOCIATED WITH EACH MOTION AXIS SEGMENT.

S U P E R I O R   R O U T I N E S

MOTAXI - GENERATE AN INTERPOLATION INFORMATION ARRAY FOR A MOTION AXIS SYSTEM

BEAMI - GENERATE AN INTERPOLATION INFORMATION ARRAY FOR A MULTIPLE BEAM MOTION AXIS LIKE SYSTEM.

D E S C R I P T I O N

IF THERE ARE TWO DEFINITION POINTS THE EQUATION OF A STRAIGHT LINE IS GENERATED, OTHERWISE THE SLOPE OF THE MOTION AXIS IS DETERMINED AT DEFINITION POINTS, AND THE EQUATIONS OF A CUBIC ARE GENERATED FOR EACH SEGMENT. THE SLOPES ARE DETERMINED BY SOLVING A SET OF SIMULTANEOUS EQUATIONS FOR THE MINIMUM INTEGRAL OF THE SECOND DERIVATIVE. THE ARC LENGTH TO EACH MOTION AXIS DEFINITION POINT IS CALCULATED, AND THE MAPPING POINT FOR EACH SEGMENT CALCULATED. THE MAPPING POINT, USED TO MAP ANY ARBITRARY POINT IN THE SEGMENT REGION TO A REFERENCE POINT ON THE MOTION AXIS, IS DEFINED TO BE THE INTERSECTION OF THE REFERENCE LINES ON EITHER END OF THE SEGMENT.

MOTAXI

P U R P O S E

PERFORM AN INTERPOLATION IN TWO VARIABLES USING AS THE INTERPOLATING FUNCTION A CUBIC SPLINE IN ARC LENGTH FOR DISPLACEMENTS AND ROTATIONS ALONG A CONTINUOUS PLANAR CURVE WITH SPECIFIED MAPPING FROM THE CURVE TO OTHER POINTS IN THE PLANE.

GIVEN A SET OF POINTS WITH ASSOCIATED REFERENCE LINES DEFINING A MOTION AXIS, AND A NUMBER OF MOTION STATIONS WITH DISPLACEMENTS AND ROTATIONS FOR A NUMBER OF MODES, GENERATE THE FUNCTIONAL DESCRIPTION OF THE MOTION AXIS, CALCULATE THE DERIVATIVES W.R.T. ARC LENGTH OF THE DISPLACEMENTS, AND FORM AN INTERPOLATION INFORMATION ARRAY CONTAINING THESE DATA.

D E S C R I P T I O N

THE MOTION AXIS FUNCTIONAL DEFINITION IS A SERIES OF CUBICS, X = C(Y), WHICH MINIMIZE THE INTEGRAL OF THE SECOND DERIVATIVE OF X WITH RESPECT TO Y. THE MOTION AXIS PASSES THROUGH THE DEFINITION POINTS THE SECOND DERIVATIVES ARE ZERO AT EITHER END. THE REFERENCE LINES THROUGH THE DEFINITION POINTS DIVIDE THE X,Y PLANE INTO REGIONS ASSOCIATED WITH EACH MOTION AXIS SEGMENT. THE REFERENCE LINES ARE ASSUMED TO BE RIGID WITH RESPECT TO DISPLACEMENTS AND ROTATIONS OF THEIR DEFINITION POINT. EACH MOTION AXIS SEGMENT HAS ASSOCIATED WITH IT A MAPPING POINT (THE INTERSECTION OF THE REFERENCE LINES ON EITHER END) OR SLOPE (THE SLOPE OF PARALLEL REFERENCE LINES ON EITHER END).

ANY ARBITRARY POINT (X,Y) IS ASSOCIATED WITH ONE MOTION AXIS SEGMENT. THE REFERENCE POINT ON THE SEGMENT IS THE INTERSECTION OF A REFERENCE LINE THROUGH (X,Y) AND THE MOTION AXIS. THE REFERENCE LINE IS THE LINE THROUGH (X,Y) AND THE SEGMENT MAPPING POINT, OR THE LINE THROUGH (X,Y) WITH THE SEGMENT MAPPING SLOPE.

THE MOTION AXIS DEFINITION EQUATIONS AND MAPPING DATA IS CALCULATED AND TOGETHER WITH DEFINITION POINTS AND REFERENCE LINES PLACED IN THE INTERPOLATION INFORMATION ARRAY. THE REFERENCE POINT IS DETERMINED FOR EACH MOTION STATION, AND THE DISPLACEMENTS AND ROTATIONS TRANSFERRED TO THE REFERENCE POINTS. THE DERIVATIVES WITH RESPECT TO ARC LENGTH OF THE DISPLACEMENTS AND ROTATIONS ARE DETERMINED (AGAIN A MINIMIZATION OF THE SECOND DERIVATIVE), AND THE DISPLACEMENTS. ROTATIONS, AND THEIR DERIVATIVES ARE PLACED IN THE INTERPOLATION INFORMATION ARRAY.

THE MODAL INFORMATION AT THE MOTION STATIONS MAY BE ORIENTED IN THE X,Y AXES (NORMAL) OR THE USER MAY INDICATE (REFER TO INDC) THAT RX IS PERPENDICULAR TO THE STRAIGHT LINE SEGMENT BETWEEN THE DEFINITION POINTS OF THE REFERENCE MOTION AXIS SEGMENT FOR THE MOTION STATION.

MOTAXO

P U R P O S E

PERFORM AN INTERPOLATION IN TWO VARIABLES USING AS THE INTERPOLATING FUNCTION A CUBIC SPLINE IN ARC LENGTH FOR DISPLACEMENTS AND ROTATIONS ALONG A CONTINUOUS PLANAR CURVE WITH SPECIFIED MAPPING FROM THE CURVE TO OTHER POINTS IN THE PLANE.

GIVEN AN INTERPOLATION INFORMATION ARRAY FOR A MOTION AXIS SYSTEM GENERATED IN ROUTINE MOTAXI, CALCULATE MODAL DISPLACEMENTS AND OPTIONALLY SLOPES AT A SET OF OUTPUT POINTS

D E S C R I P T I O N

CONTAINED WITHIN THE INTERPOLATION INFORMATION ARRAY IS THE DEFINITION OF THE MOTION AXIS AS A SERIES OF CUBICS IN Y WITH MAPPING ON EACH SEGMENT SPECIFIED BY A MAPPING POINT. THE MODAL DISPLACEMENTS (TRANSLATIONS AND ROTATIONS) OF POINTS ON THE MOTION AXIS AND THE DERIVATIVES OF DISPLACEMENT WITH RESPECT TO ARC LENGTH ALSO EXIST WITHIN THE ARRAY.

FOR EACH OUTPUT POINT, THE ASSOCIATED MOTION AXIS REFERENCE POINT IS DETERMINED, AND THE MODAL DISPLACEMENTS AT THE REFERENCE POINT CALCULATED USING HERMITE INTERPOLATION. THE MODAL DISPLACEMENTS AT THE OUTPUT POINT ARE THEN

```
TZ(X,Y) = TZ(S) + DR&-SIN(THETA)*RX(S) + COS(THETA)*RY(S)&

D(TZ)/DX = -RY(S) + DR&-SIN(T)*D(RX)/D(S)+COS(T)*D(RY)/D(S)&D(S)/D(X)
D(TZ)/DY =  RX(S) + DR&-SIN(T)*D(RX)/D(S)+COS(T)*D(RY)/D(S)&D(S)/D(Y)
```

WHERE THETA(T) IS THE ANGLE OF THE REFERENCE LINE FROM X AXIS, DR IS THE DISTANCE FROM OUTPUT POINT TO REFERENCE POINT, AND S IS THE ARC LENGTH ALONG THE MOTION AXIS TO THE REFERENCE POINT.

MOTPTI

P U R P O S E

TO PERFORM A TRANSFORMATION OF MOTIONS FROM A SINGLE POINT

D E S C R I P T I O N

THE MOTION OF A POINT IN 3-SPACE IS DEFINED BY A RIGID TRANSFORMATION APPLIED TO THE MOTION OF A DATUM

MOTPTO

P U R P O S E

MOTPTO CALCULATE MODAL DISPLACEMENTS AND OPTIONALLY SLOPES AT A SET OF OUTPUT AERODYNAMIC CONTROL POINTS, GIVEN AN INTERPOLATION INFORMATION ARRAY (SA ARRAY) GENERATED IN ROUTINE MOTPTI FOR A MOTION POINT SYSTEM

D E S C R I P T I O N

MOTPTO PERFORMS ITS TASKS IN THE FOLLOWING STEPS
FOR EACH OUTPUT POINT SPECIFIED, THE ROUTINE GENERATES DISPLACEMENTS AND (OPTIONALLY) SLOPES BY A RIGID TRANSFORMATION OF THE MOTION OF A DATUM.

NAMFIL

P U R P O S E

NAMFIL CONVERTS A HOLLERITH WORD INTO A LEFT JUSTIFIED ZERO FILLED WORD SUITABLE FOR USE AS A FILE NAME. THE INPUT FILE NAME MUST ...
  (1) BEGIN WITH AN ALPHABETIC CHARACTER, A TO Z.
  (2) BE LEFT JUSTIFIED (NO LEADING BLANKS).
  (3) CONTAIN FROM 1 TO 7 CHARACTERS PRIOR TO A BLANK.
  (4) END WITH FROM 3 TO 9 BLANKS.
OR AN ERROR IS DIAGNOSED BY SETTING THE VALUE OF THE FUNCTION TO ZERO.

D E S C R I P T I O N

NAMFIL PERFORMS ITS TASK IN THE FOLLOWING STEPS.
  (1) INITIALIZE THE VALUE OF THE FUNCTION TO ZERO AND THEN CHECK FOR A BLANK NAME OR ONE NOT BEGINNING WITH AN ALPHA. CHARACTER. JUMP TO (6) IF THERE IS AN ERROR.
  (2) STORE THE HOLLERITH NAME IN ITEMP.
  (3) CHECK THE LAST CHARACTER OF ITEMP.
      IF IT IS NOT A BLANK JUMP OT (6).
      IF IT IS A BLANK CHANGE IT TO BINARY ZERO, AND THEN CIRCULAR SHIFT ITEMP ONE CHARACTER TO THE RIGHT. STEP (3) IS REPEATED UNTIL ALL THEN CHARACTERS HAVE BEEN PROCESSED OR A NON-BLANK CHARACTER IS FOUND.
  (4) IF THE NUMBER OF NON-BLANK CHARACTERS IS 0 OR GREATER THAN 7 JUMP TO (6).
  (5) LEFT JUSTIFY ITEMP AND SET THE FUNCTION VALUE EQUAL TO ITEMP.
  (6) RETURN TO THE CALLING PROGRAM.

ORDER

P U R P O S E

TO REORDER THE VALUES OF AN ARRAY MONOTONICALLY INCREASING AND THE RELATED INDEX ACCORDINGLY

D E S C R I P T I O N

ORDER PLACES X IN MONOTONIC INCREASING ORDER AND REORDERS THE INDEX SET ACCORDINGLY

PLATEA

P U R P O S E

PLATEA FORMS THE PLATE MATRIX COEFFICIENT OF SYSTEM OF EQUATIONS

S U P E R I O R   R O U T I N E S
---------------   ---------------

PLATEA IS CALLED BY PLATFI

D E S C R I P T I O N

FORM THE COEFFIEIENT MATRIX FOR SYSTEM OF EQUATION, SK = SMOOTHING CONSTANT (RATIO OF PLATE STIFFNESS TO INPUT POINT SPRING STIFFNESS)

```
                   * 1 X(1) Y(1)      WHERE A(I,J) = R**2 LN(R**2), I&J
                   * -   -    -
      A(I,J)       * -   -    -                    = 0, I=J     INDS = 0
                   * -   -    -
                   * 1 X(N) Y(N)                   = SK(1), I=J    INDS = 1
                   *
*****************************                      = SK(I), I=J    INDS = 2
                   *
  1  ---    1      * C    C     C              R**2 = (X(I)-X(J))**2
X(1) ---  X(N)     * C    C     C                    +(Y(I)-Y(J))**2
Y(1) ---  Y(N)     * C    0     C              N=NO. PTS., M=N+3=NO. EQUATIONS
```

PLATEI

P U R P O S E

TO PERFORM A BIVARIATE INTERPOLATION USING AS THE INTERPOLATING FUNCTION THE SMALL DEFLECTION EQUATION OF AN INFINITE PINNED PLATE

D E S C R I P T I O N

THE FUNCTION TO BE INTERPOLATED IS APPROXIMATED BY

```
F(X,Y) = SUM& A(K)*R(K)**2 * LN(R(K)**2), K=1,N &
         + A(N+1) + A(N+2)*X + A(N+3)*Y
```

WHERE R(K)**2 = (X-X(K))**2 + (Y-Y(K))**2, (X(K),Y(K))= INPUT POINTS, A(K) = INTERPOLATING FUNCTION COEFFICIENTS. THE N+3 UNKNOWNS, A(K), ARE FOUND BY SOLVING THE LINEAR SYSTEM OF N+3 EQUATIONS,

```
F(X(K),Y(K)) = Z(K), K=1,N
SUM& A(K), K=1,N & = 0
SUM& A(K)*X(K), K=1,N & = 0
SUM& A(K)*Y(K), K=1,N & = 0
```

THE EQUATION MAY BE ANALYTICALLY DIFFERENTIATED TO GIVE

```
DF(X,Y)/DX = SUM& 2*A(K)*(X-X(K))*(LN(R(K)**2) + 1), K=1,N&
             + A(N+2)

DF(X,Y)/DY = SUM& 2*A(K)*(Y-Y(K))*(LN(R(K)**2) + 1), K=1,N&
             + A(N+3)
```

FOR SITUATIONS IN WHICH SMOOTHING IS DESIRABLE, THE PINNED CONSTRAINT, F(X(K),Y(K)) = Z(K), MAY BE REMOVED AND AN ARTIFICIAL SPRING PLACED AT THE INPUT POINT, IN WHICH CASE

```
F(X(K),Y(K)) = SUM& A(K)*R(K)**2*LN(R(K)**2)+A(K)*S(K),K=1,N&
               + A(N+1) + A(N+2)*X(K) + A(N+3)*Y(K)
```

WHERE S(K) IS THE RATIO OF PLATE STIFFNESS TO SPRING STIFF-NESS AT THE K-TH INPUT POINT. NOTE THAT S(K)=0 IS EQUIVALENT TO PINNED, S(K)=1 IS APPPROXIMATELY 70 PERCENT SMOOTHING, S(K)= INFINITE FOR ALL K WILL GIVE APPROXIMATELY THE LEAST SQUARES PLANE.

UNLESS SPECIFICALLY REQUESTED OTHERWISE A TRANSFORMATION AND SCALING OF COORDINATES IS PERFORMED WHICH CENTERS THE DATA POINTS, ROTATES TO PRINCIPLE AXES, AND DIVIDES THE RESULTING COORDINATES BY THE RADII OF GYRATION

PLATEO

P U R P O S E

PLATEO CALCULATE MODAL DISPLACEMENTS AND OPTIONALLY SLOPES AT A SET OF OUTPUT AERODYNAMIC CONTROL POINTS, GIVEN AN INTERPOLATION INFORMATION ARRAY (SA ARRAY) GENERATED IN ROUTINE PLATEI FOR A PLATE SYSTEM

D E S C R I P T I O N

PLATEO CALCULATE THE VALUES OF THE FUNCTIONS AND THEIR DERIVATIVES AT A SET OF OUTPUT POINTS FROM THE GIVEN SPLINE COEFFICIENTS FOR A SET OF FUNCTIONS AS DETERMINED IN ROUTINE PLATEI AND THE ASSOCIATED INPUT POINTS.
THE FUNCTION TO BE INTERPOLATED IS APPROXIMATED BY

```
F(X,Y) = SUM& A(K)*R(K)**2 * LN(R(K)**2), K=1,N &
         + A(N+1) + A(N+2)*X + A(N+3)*Y
```

WHERE R(K)**2 = (X-X(K))**2 + (Y-Y(K))**2, (X(K),Y(K))= INPUT POINTS, A(K) = INTERPOLATING FUNCTION COEFFICIENTS. THE N+3 UNKNOWNS, A(K), ARE FOUND BY SOLVING THE LINEAR SYSTEM OF N+3 EQUATIONS,

```
            F(X(K),Y(K)) = Z(K), K=1,N
            SUM& A(K), K=1,N & = 0
            SUM& A(K)*X(K), K=1,N & = 0
            SUM& A(K)*Y(K), K=1,N & = 0
```

THE EQUATION MAY BE ANALYTICALLY DIFFERENTIATED TO GIVE

```
DF(X,Y)/DX = SUM& 2*A(K)*(X-X(K))*(LN(R(K)**2) + 1), K=1,N&
             + A(N+2)

DF(X,Y)/DY = SUM& 2*A(K)*(Y-Y(K))*(LN(R(K)**2) + 1), K=1,N&
             + A(N+3)
```

FOR SITUATIONS IN WHICH SMOOTHING IS DESIRABLE, THE PINNED CONSTRAINT, F(X(K),Y(K)) = Z(K), MAY BE REMOVED AND AN ARTIFICIAL SPRING PLACED AT THE INPUT POINT, IN WHICH CASE

```
F(X(K),Y(K)) = SUM& A(K)*R(K)**2*LN(R(K)**2)+A(K)*S(K),K=1,N&
               + A(N+1) + A(N+2)*X(K) + A(N+3)*Y(K)
```

WHERE S(K) IS THE RATIO OF PLATE STIFFNESS TO SPRING STIFFNESS AT THE K-TH INPUT POINT. NOTE THAT S(K)=0 IS EQUIVALENT TO PINNED, S(K)=1 IS APPROXIMATELY 70 PERCENT SMOOTHING, S(K)= INFINITE FOR ALL K WILL GIVE APPROXIMATELY THE LEAST SQUARES PLANE.

UNLESS SPECIFICALLY REQUESTED OTHERWISE A TRANSFORMATION AND SCALING OF COORDINATES IS PERFORMED WHICH CENTERS THE DATA POINTS, ROTATES TO PRINCIPLE AXES, AND DIVIDES THE RESULTING COORDINATES BY THE RADII OF GYRATION

PLATET

P U R P O S E

PLATET PERFORMS A TRANSFORMATION OF COORDINATES FROM LOCAL (X,Y) AXIS TO PRINCIPLE (U,V) AXIS OR FROM PRINCIPLE (U,V) AXIS TO LOCAL (X,Y) AXIS

S U P E R I O R   R O U T I N E S
---------------   ---------------

PLATET IS CALLED BY PLATEI

D E S C R I P T I O N

PERFORM A TRANSFORMATION OF COORDINATES FROM (X,Y) TO (U,V) IF NIND 0, OR FROM (U,V) TO (X,Y) IF NIND 0.

```
    *U*   *1/RGU    0   * * COST SINT* *X-XBAR*
    * * = *             * *        * *      *
    *V*   *  0    1/RGV* *-SINT COST* *Y-YBAR*
```

WHERE COST = COS(THETA), SINT = SIN(THETA), THETA IS THAT ANGLE SUCH THAT PUV = SUM&U(I)*V(I)&/N = 0, RGU,RGV = RADII OF GYRATION IN (U,V), XBAR,YBAR = C.G. LOCATION IN (X,Y) - NOTE UBAR,VBAR = 0,0

POLYI

P U R P O S E

POLYI GENERATE THE INTERPOLATION INFORMATION ARRAY (SA ARRAY) FOR THE POLYNOMIAL METHOD OF INTERPOLATION

D E S C R I P T I O N

POLYI PERFORM ITS TASKS IN THE FOLLOWING STEPS

(1) LOAD THE FIRST 7 WORDS, PARAMETERS FROM ARGUMENT LIST, INTO THE SA ARRAY
(2) LOAD THE POLYNOMIAL COEFFICIENTS INTO THE SA ARRAY
(3) RETURN TO CALLING ROUTINE

POLYO

P U R P C S E

POLYO CALCULATE MCDAL DISPLACEMENTS AND OPTICNALLY SLCPES AT A SET CF
OUTPUT AERODYNAMIC CONTRCL POINTS, GIVEN AN INTERPCLATION INFCRMATICN
ARRAY (SA ARRAY) GENERATEC IN ROUTINE PCLYI FOR A PCLYNCMIAL SYSTEM

C E S C R I P T I C N

POLYC PERFORMS ITS TASKS IN THE FOLLCWING STEPS
  (1) INITIALIZATION
      (A) EXTRACT CONSTANTS FRCM SA ARRAY
      (B) CHECK VALUES OF PARAMETERS IF WITHIN LIMITS
  (2) LCCP ON NUMBER CF MCDES
  (3) SET LEADING CR TRAILING COLUMNS TC ZERC IF REQUIREC
  (4) SETUP POLYNOMIAL COEFFICIENTS FCR ITH MODE
  (5) FCR EACH PCINT COMPUTE DISPLACEMENT AND CPTICNALLY SLCPES
  (6) GC TC (2) FOR ANOTHER MODE
  (7) RETURN TC CALLING RCUTINE

PRGBEG

P U R P O S E

PRGBEG PRINTS A ONE PAGE HEADER CONTAINING THE NAME AND VERSICN OF
THE PROGRAM BEING RUN AND THE CURRENT DATE AND TIME OF EXECUTICN.

C E S C R I P T I O N

PRGBEG CETERMINES THE CURRENT DATE AND TIME AND THE WRITES A
HEADER MESSAGE CN THE PRINTED OUTPUT FILE.

PRGEND

P U R P C S E

PRGEND PRINTS A ONE PAGE HEADER CONTAINING THE NAME AND VERSICN OF
THE PRCGRAM BEING RUN AND THE CURRENT DATE AND TIME OF EXECUTICN.

C E S C R I P T I C N

PRGEND CETERMINES THE CURRENT DATE AND TIME AND THE WRITES A
ENDING MESSAGE CN THE PRINTED OUTPUT FILE.

PRINTM

P U R P O S E

PRINTM WRITES A TWO DIMENSIONAL ARRAY (MATRIX) AND ITS ASSOCIATED TEXTUAL DESCRIPTION ON THE BCD FILE IUTFIL. NORMALLY, IUTFIL IS THE PRINTED OUTPUT FILE. THE MATRIX ELEMENTS WILL BE PRECEDED BY THE MATRIX LABEL AND SIZE.

NOTE - THE MATRIX MUST BE REAL (CONTAINING FLOATING POINT ELEMENTS).

D E S C R I P T I O N

PRINTM PERFORMS ITS TASK IN THE FOLLOWING STEPS
- (1) INITIALIZE THE ERROR CODE.
- (2) WRITE THE MATRIX LABEL AND SIZE ON IUTFIL.
  JUMP TO (4) IF THE NUMBER OF WORDS IN LABEL IS ILLEGAL.
  JUMP TO (4) IF THE MATRIX SIZE IS ILLEGAL.
- (3) WRITE THE MATRIX ELEMENTS ON IUTFIL BY ROWS.
- (4) RETURN TO CALLING PROGRAM.

PRINTCM

P U R P O S E

PRNTCM WRITES A TWO DIMENSIONAL ARRAY (MATRIX) AND ITS ASSOCIATED TEXTUAL DESCRIPTION ON THE BCD FILE IUTFIL. NORMALLY, IUTFIL IS THE PRINTED OUTPUT FILE. THE MATRIX ELEMENTS WILL BE PRECEDED BY THE MATRIX LABEL AND SIZE.

NOTE - THE MATRIX MUST BE REAL (CONTAINING FLOATING POINT ELEMENTS).

D E S C R I P T I O N

PRNTCM PERFORMS ITS TASK IN THE FOLLOWING STEPS
- (1) INITIALIZE THE ERROR CODE.
- (2) WRITE THE MATRIX LABEL AND SIZE ON IUTFIL.
  JUMP TO (4) IF THE NUMBER OF WORDS IN LABEL IS ILLEGAL.
  JUMP TO (4) IF THE MATRIX SIZE IS ILLEGAL.
- (3) WRITE THE MATRIX ELEMENTS ON IUTFIL BY ROWS.
- (4) RETURN TO CALLING PROGRAM.

PUTFIT

P U R P O S E

PUTFIT ADDS A FILE NAME TO THE RA+2 LIST.

S U P E R I O R   R O U T I N E S

NAME	ORIGIN	DESCRIPTION
FETAC	DYLIB	SET UP A FILE FOR USE IN FORTRAN PROGRAM - THE FIT, FET, AND THE RA+2 ENTRY.
FETACC	DYLIB	SET UP A FILE FOR USE IN FORTRAN PROGRAM - THE FIT, FET, AND THE RA+2 ENTRY.

D E S C R I P T I O N

PUTFIT EXTRACTS THE FILE NAME, LFN, FROM THE FIT AND SEARCHES FOR THAT FILE NAME IN THE RA+2 LIST. IF LFN IS IN THE LIST AN ERROR CODE OF -2 IS RETURNED. IF THE LIST IS ALREADY FULL (MAX. 49) AN ERROR CODE OF -1 IS RETURNED. OTHERWISE, THE FILE NAME AND FIT ADDRESS ARE PLACED IN THE FIRST ZERO WORD OF THE LIST.

READTP

```
P U R P O S E
-------------

READTP READS  A TWO DIMENSIONAL ARRAY (MATRIX) FROM A SEQUENTIAL
MAGNETIC FILE.  THE INFORMATION IS OBTAINED IN TWO LOGICAL RECORDS
WITH FORTRAN BINARY READ STATEMENTS.

D E S C R I P T I O N
---------------------

READTP PERFORMS ITS TASK IN THE FOLLOWING STEPS.
  (1) INITIALIZE THE ERROR CODE AND CHECK THE INPUT ARGUMENTS FOR
      ERRORS.
      JUMP TO (5) IF THERE ARE ERRORS.
  (2) POSITION THE FILE AS REQUIRED AFTER CHECKING THE SPACING
      PARAMETERS FOR ERRORS.
      JUMP TO (5) IF SPACING PARAMETERS ARE ILLEGAL.
      JUMP TO (5) IF SPACING OF FILES OR MATRICES FAILS.
  (3) READ THE FIRST RECORD.
      IT WILL CONTAIN
         WORD
            1   NAME
            2   NROWS
            3   NCOLS
            4   0
            5   ITYPE
            6   NELEM
         7-10   0
        11-16   OPTIONAL USER DATA PASSED THROUGH THE FIRST RECORD IF
                STORED IN WORDS 1-5 OF THE AUXILIARY ID GIVEN TO WRTETP
                WHEN THE MATRIX WAS WRITTEN ONTO IFILE.
      JUMP TO (5) IF AN END-OF-FILE WAS READ.
      JUMP TO (5) IF THE NAME DOES NOT MATCH THE SPECIFIED ONE.
      JUMP TO (5) IF NROWS .GT. NROWD.
  (4) READ THE SECOND RECORD.
      IF ITYPE=6HM=NULL,  THE SECOND RECORD IS READ BUT NOT STORED
                          AND MATRIX IS FILLED WITH ZEROS.
      IF ITYPE=6HSPARSE,  THE SECOND RECORD OF NWORD IS READ INTO THE
                          BUFFER ARRAY BUFF.  THEN, EVERY ELEMENT WHICH
                          IS AN INTEGER IS REPLACED BY THAT NUMBER OF
                          ZEROS.  FINALLY, THE ARRAY ELEMENTS ARE
                          EXTRACTED FROM BUFF AND TRANSPOSED INTO THE
                          OUTPUT ARRAY MATRIX.
      IF ITYPE=0,         THE SECOND RECORD IS READ DIRECTLY INTO
                          MATRIX BY ROWS.
      JUMP TO (5) IF AN END-OF-FILE WAS READ.
  (5) RETURN
```

REFPT

P U R P O S E

GIVEN A POINT, (X,Y), THE CUBIC DEFINITION OF A MOTION AXIS SEGMENT, AND EITHER A MAPPING POINT, (XM,YM), OR THE SLOPE OF A REFERENCE LINE, DYDXRL, DETERMINE THE REFERENCE POINT DEFINED TO BE THE INTERSECTION OF THE REFERENCE LINE THROUGH (X,Y) AND THE MOTION AXIS.

S U P E R I O R   R O U T I N E S

MAATTCH - DETERMINE MOTION AXIS REFERENCE POINT INFORMATION

D E S C R I P T I O N

THE EQUATION OF THE REFERENCE LINE IS DETERMINED FROM (X,Y) AND (XM,YM) OR (X,Y) AND SLOPE DYDXRL IF THE MAPPING POINT IS INDEFINITE. SOLVING

```
        CC + C1*YR + C2*YR**2 + C3*YR**3 = XR
                           DX*(YR-Y) + X = XR
```

GIVES

```
        CCM + C1M*YR + C2*YR**2 + C3*YR**3 = 0
```

WHICH HAS THREE, TWO, OR ONE ROOTS DEPENDENT UPON C3 AND C2. THE ROOT ON THE INTERVAL (YMA1,YMA2) IS SELECTED, AND XR CALCULATED FROM THE CUBIC. IF NO ROOT FALLS IN THE INTERVAL, THE CLOSEST ENDPOINT IS USED, UNLESS YMA1=YMA2 AND C3=C2=0 (I.E. EXTRAPOLATION), IN WHICH CASE THE SINGLE ROOT IS ACCEPTED.

RETURNF

P U R P O S E

RETURNF ALLOWS A FORTRAN PROGRAM TO ELIMINATE A TEMPORARY STORAGE FILE. THE FILE NAME WILL NO LONGER EXIST AND THE DISK SPACE WILL BE RELEASED FOR OTHER USE.

D E S C R I P T I O N

RETURNF CHECKS TO SEE THAT THE FILE NAME IS LEGAL AND THEN CALLS EVICT TO ACCOMPLISH ITS PURPOSE.

RQL/IRQL

P U R P O S E

RQL IS CALLED TO SHIFT THE BITS OF WORD NBITS POSITIONS.
THE SHIFT IS LEFT CIRCULAR IF NBITS IS POSITIVE.
THE SHIFT IS TO THE RIGHT WITH SIGN EXTENSION AND END OFF IF NBITS IS NEGATIVE.

NOTE ...THE RESULTS ARE UNDEFINED IF NBITS IS NOT BETWEEN 0 AND 60.

A SECOND ENTRY POINT, IRQL, IS PROVIDED TO ALLOW EITHER REAL OR INTEGER MODE IN ARITHMETIC STATEMENTS.

D E S C R I P T I O N

IRQL PERFORMS ITS TASK BY CALLING SHIFT.

SEARCH

P U R P O S E

PERFORM A BINARY TABLE SEARCH.

D E S C R I P T I O N

A BINARY SEARCH IS PERFORMED ON A MONOTONICALLY ORDERED ARRAY X TO FIND NO POINTS THAT CENTER AROUND Z.

STARTR

P U R P O S E

STARTR IS CALLED TO INITIALIZE THE VARDIM STORAGE SCHEME.

D E S C R I P T I O N

STARTR MUST BE CALLED BEFORE ANY OTHER VARDIM ROUTINE. IT CREATES THE ARRAY CATALOGUE, SETS THE MAXIMUM NUMBER OF ARRAYS, AND DETERMINES THE MAXIMUM ARRAY STORAGE AVAILABLE.

******* THE VARDIM VARIABLE DIMENSION STORAGE SCHEME ****************

VARDIM IS A SET OF 5 SUBROUTINES WHICH HANDLES THE STORAGE OF ARRAYS DEFINED DURING PROGRAM EXECUTION, I.E VARIABLE DIMENSIONING OR DYNAMIC CORE ALLOCATION.

VARDIM USES BLANK COMMON FOR ALL ARRAY STORAGE. THIS METHOD IS POSSIBLE ON THE CDC 6600 WHICH PLACES BLANK COMMON AT THE END OF ALL OTHER PROGRAM STORAGE. THE USER MUST REQUEST ENOUGH FIELD LENGTH TO PROVIDE SUFFICIENT ARRAY STORAGE BETWEEN THE BEGINNING OF BLANK COMMON AND THE END OF HIS FIELD LENGTH.

------- GENERAL COMMENTS

1) ARRAY STORAGE MAY BE VARIABLE WITHIN A SINGLE PROGRAM RUN AS WELL AS BETWEEN RUNS. ARRAYS MAY BE DEFINED OR DELETED AT ANY TIME. IT IS NOT NECESSARY TO DEFINE THEM ALL AT THE BEGINNING OF THE PROGRAM.

2) EACH ARRAY WILL BE IDENTIFIED BY A SIX CHARACTER HOLLERITH NAME (LEFT JUSTIFIED AND BLANK FILLED), MAY HAVE FROM ONE TO THREE DIMENSIONS, AND MAY BE TYPED REAL OR INTEGER. THE TYPE IS CRITICAL ONLY IF THE SUBROUTINE PRINTR WILL BE CALLED (SEE PRINTR FOR USE OF VARIABLE INTREL).

3) NEWLY DEFINED ARRAYS WILL BE NULL.

4) THE ARRAY STORAGE IS ALWAYS COMPACTED TO USE THE FIRST WORDS OF BLANK COMMON. A NEWLY DEFINED ARRAY IS ALWAYS LOCATED AFTER PRE-EXISTING ARRAYS. IF ARRAY I IS DELETED THEN ARRAYS I+1 THROUGH N WILL BE MOVED FORWARD TO POSITIONS I THRU N-1.

5) WHEN ANY VARDIM ROUTINE IS CALLED IT MAKES CHECKS TO SEE THAT
   A) DUPLICATE ARRAY NAMES ARE NOT DEFINED
   B) ILLEGAL DIMENSIONS ARE NOT SPECIFIED (.LE.0 IS ILLEGAL)
   C) THE ARRAY CATLOGUE HAS NOT BEEN EXCEEDED
   D) THE AVAILABLE CORE HAS NOT BEEN EXCEEDED

STARTR (Continued)

```
THE BOOKKEEPING PERFORMED BY VARDIM IS STORED IN A CATALOGUE
ARRAY, KATLOG, AND THE LABELED COMMON BLOCK /VARDIM/.

6)  THE VARDIM ARRAY CATALOG, KATLOG, IS ITSELF AN ARRAY IN
    VARDIM STORAGE WITH THE DIMENSIONS (6,NMAX,1).  EACH ARRAY
    WILL HAVE A SIX WORD ENTRY IN KATLOG (I.E. ONE COLUMN).
    THE SIX WORDS CONTAIN
    A) NAME     - 6 HOLLERITH CHARACTERS (LEFT JUSTIFIED AND
                  BLANK FILLED)
    B) LOCATION - FIRST WORD ADDRESS OF THE ARRAY IN BLANK COMMON
    C) TYPE     - 0 FOR INTEGER AND 1 FOR REAL
    D) FIRST  DIMENSION OF THE ARRAY
    E) SECOND DIMENSION OF THE ARRAY
    F) THIRD  DIMENSION OF THE ARRAY

7)  ALL VARDIM ROUTINES EXCEPT XFER CONTAIN THE LABELED COMMON
    BLOCK /VARDIM/ WHICH HAS THE BOOKKEEPING VARIABLES
    A) NMAX     - MAXIMUM NUMBER OF ARRAY POSSIBLE ON THIS PROGRAM
                  RUN (DEFINED WHEN SUBROUTINE STARTR IS CALLED)
    B) NENTRY   - NUMBER OF ARRAYS CURRENTLY DEFINED
    C) LWAVAL   - LAST WORD AVAILABLE IN BLANK COMMON
    D) LWUSED   - LAST WORD CURRENTLY IN USE IN BLANK COMMON
    E) LKAT     - FIRST WORD ADDRESS OF KATLOG IN BLANK COMMON
    F) MAXUSD   - MAXIMUM CORE ADDRESS EVER USED BY VARDIM
                  STORAGE.

8)  ALL VARDIM ROUTINES EXCEPT XFER CONTAIN THE BLANK COMMON
    DEFINITION
          COMMON D(1)
          DIMENSION ID(1), KATLOG(6,1)
          EQUIVALENCE (D,ID,KATLOG)

9)  AS OF NOV. 16, 1972,  THE VARDIM ROUTINES REQUIRE LESS THAN
    2200 (OCTAL) WORDS OF CORE EXCLUSIVE OF BLANK COMMON.
```

```
------- SHORT DESCRIPTIONS OF ALL VARDIM SUBROUTINES

    STARTR
        STARTR IS CALLED TO INITIALIZE THE VARDIM STORAGE SCHEME.
        STARTR MUST BE CALLED BEFORE ANY OTHER VARDIM ROUTINE.  IT
        CREATES THE ARRAY CATALOGUE, SETS THE MAXIMUM NUMBER OF
        ARRAYS, AND DETERMINES THE MAXIMUM ARRAY STORAGE AVAILABLE.

    INITIR
        INITIR IS CALLED TO INITIALIZE (DEFINE) A NEW ARRAY.  IT MUST
        BE CALLED BEFORE ANY OTHER VARDIM ROUTINE REFERS TO THE ARRAY.
        INITIR ALLOCATES THE ARRAY STORAGE, ZEROS THE AREA, AND MAKES
        AN ENTRY IN THE ARRAY CATALOGUE.  IF THE ARRAY BEING DEFINED
        ALREADY EXISTS, THE ELEMENTS ARE SIMPLY SET TO ZERO IF THE
        DIMENSIONS ARE TO REMAIN THE SAME.  IF THE ARRAY SIZE IS TO
        BE CHANGED THE OLD ARRAY IS DELETED AND A NEW ONE DEFINED.

    DELETR
        DELETR IS CALLED TO DELETE OR ELIMINATE AN ARRAY FROM
        VARDIM STORAGE.  THE ARRAYS ENTRY IN THE CATALOGUE WILL
        DISAPPEAR AND BOTH THE CATALOGUE AND ARRAY STORAGE WILL BE
        COMPRESSED.

    LOCATR
        LOCATR IS CALLED TO DETERMINE AN ARRAYS SIZE, TYPE, AND
        LOCATION.  LOCATR SHOULD BE CALLED JUST PRIOR TO HANDLING
        THE ARRAY BECAUSE ITS LOCATION CHANGES AS ARRAYS ARE DELETED
        AND ADDED TO VARDIM.

    XFER
        XFER IS A HIDDEN ROUTINE IN THE VARDIM STORAGE SCHEME.
        ROUTINE DELETR WILL EXECUTE XFER TO QUICKLY MOVE BLOCKS OF
        CORE.
```

TBLU1

P U R P O S E

PERFORM TABLE SEARCH AND LAGRANGIAN POLYNOMIAL INTERPOLATION OF USER-DEFINED DEGREE ON ONE INDEPENDENT VARIABLE.

D E S C R I P T I O N

TBLU1 USES A BINARY SEARCH TECHNIQUE TO LOCATE THE PROPER POSITION IN A TABLE AND THEN USES A LAGRANGIAN INTERPOLATING POLYNOMIAL OF USER-DEFINED DEGREE.

TERP1

P U R P O S E

POLYNOMIAL INTERPOLATION FOR ONE INDEPENDENT VARIABLE

D E S C R I P T I O N

A LAGRANGIAN INTERPOLATING POLYNOMIAL IS EVALUATED AT THE SPECIFIED VALUES OF THE VARIABLE TO GET THE DESIRED ANSWER TERP1 WHICH COMES FROM SUM (FOR J = I THROUGH ND+I) OF THE PRODUCTS (FOR K = I THROUGH ND+I , BUT NOT FOR K = J) FOR THE EXPRESSION

```
(Z-X(K))*Y(J)/(X(J)-X(K))
```

VIP

P U R P O S E

CALCULATE C = A*B WHERE A AND B ARE REAL VECTORS.

D E S C R I P T I O N

CALCULATE REAL VECTOR INNER PRODUCT USING SINGLE PRECISION INTERNALLY.

VIPA

P U R P O S E

CALCULATE C = C + A*B WHERE A AND B ARE REAL VECTORS

D E S C R I P T I O N

CALCULATE REAL VECTOR INNER PRODUCT USING SINGLE PRECISION INTERNALLY.
ADD THE RESULT TO THE INCOMING VALUE OF C.

VIPD

P U R P O S E

CALCULATE C = A*B WHERE A AND B ARE REAL VECTORS.

D E S C R I P T I O N

CALCULATE REAL VECTOR INNER PRODUCT USING DOUBLE PRECISION INTERNALLY.
ROUND THE RESULT TO SINGLE PRECISION.

VIPDA

P U R P O S E

CALCULATE C = C + A*B WHERE A AND B ARE REAL VECTORS

D E S C R I P T I O N

CALCULATE REAL VECTOR INNER PRODUCT USING DOUBLE PRECISION INTERNALLY.
ROUND THE RESULT TO SINGLE PRECISION.
ADD THE RESULT TO THE INCOMING VALUE OF C.

VIPDS

P U R P O S E

CALCULATE   C = C - A*B   WHERE A AND B ARE REAL VECTORS.

D E S C R I P T I O N

CALCULATE REAL VECTOR INNER PRODUCT USING DOUBLE PRECISION INTERNALLY.
ROUND THE RESULT TO SINGLE PRECISION.
SUBTRACT THE RESULT FROM THE INCOMING VALUE OF C.

VIPS

P U R P O S E

CALCULATE   C = C - A*B   WHERE A AND B ARE REAL VECTORS.

D E S C R I P T I O N

CALCULATE REAL VECTOR INNER PRODUCT USING SINGLE PRECISION INTERNALLY.
SUBTRACT THE RESULT FROM THE INCOMING VALUE OF C.

VMIN

P U R P O S E

TO FIND THE MINIMUM VALUE OF AN ARRAY AND INDICATE THE
ELEMENT NUMBER OF THE MINIMUM VALUE

S U P E R I O R   R O U T I N E S

VMIN IS CALLED BY ORDER

D E S C R I P T I O N

```
VMIN RETURNS   XMIN =  MIN (   X(NSTART),  X(N)  )
ALONG WITH  THE   INDEX K, K MEASURED     FROM  1 ,
```

WRTETP

P U R P O S E

WRTETP WRITES A TWO DIMENSIONAL ARRAY (MATRIX) ON A SEQUENTIAL MAGNETIC FILE. THE INFORMATION IS WRITTEN IN TWO LOGICAL RECORDS WITH FORTRAN BINARY WRITE STATEMENTS.

D E S C R I P T I O N

```
WRTETP PERFORMS ITS TASK IN THE FOLLOWING STEPS
  (1) INITIALIZE THE ERROR CODE AND CHECK THE INPUT SIZES FOR ERRORS.
      JUMP TO (5) IF SIZES ARE ILLEGAL.
  (2) POSITION THE FILE AS REQUIRED AFTER CHECKING THE SPACING
      PARAMETERS FOR ERRORS.
      JUMP TO (5) IF SPACING PARAMETERS ARE ILLEGAL.
      JUMP TO (5) IF SPACING OF FILES OR MATRICES FAILS.
  (3) WRITE THE FIRST RECORD ONTO THE MAGNETIC FILE.
      IT WILL CONTAIN
         WORD
            1   NAME
            2   NROWS
            3   NCOLS
            4   C
            5   C
            6   NELEM= NROWS * NCOLS
         7-10   C
        11-16   AUXID(I), I=1,6
  (4) WRITE THE SECOND RECORD ONTO THE MAGNETIC FILE.
      IT WILL CONTAIN
            ((MATRIX(I,J), J=1,NCOLS), I=1,NROWS)  THE MATRIX BY ROWS
  (5) RETURN TO CALLING PROGRAM
```

XFER

D E S C R I P T I O N

SUBROUTINE XFER SIMPLY WILL MOVE THE FIRST L INTEGER VALUES OF IY INTO IX.

XPANDZ

P U R P O S E

XPANCZ IS CALLEC TO EXPAND AN ARRAY BUF OF LENGTH NWORDS TO THE LENGTH NELEM BY REPLACING ALL INTEGER ELEMENTS WITH THAT INTEGER NUMBER OF ZEROS.

S U P E R I O R   R O U T I N E S

		DESCRIPTION
READTP	DYLIB	READ A MATRIX FROM A FILE (2 RECORDS).

D E S C R I P T I O N

XPANDZ EXPANDS THE ARRAY BUFF ELEMENT BY ELEMENT FROM THE BCTTCM. EACH ELEMENT IS CHECKED TO SEE IF IT IS AN INTEGER. IF IT IS NOT AN INTEGEP IT IS SIMPLY STORED. IF IT IS AN INTEGER THAT NUMBER CF CONSECUTIVE INTEGERS IS STCRED. THE OPERATICN STCPS WHEN THE INPUT AND OUTPUT POINTERS (IA AND IB) ARE EQUAL.

ZEROCOL

P U R P O S E

ZEROCOL INITIALIZE COLUMNS NF-NL TC ZERC FCR M ROWS

S U P E R I O R   R O U T I N E S

ZEROCOL IS CALLED BY BEAMC, MOTAXC, MOTPTO, AND PLATEO

D E S C R I P T I O N

ZEROCOL INITIALIZE Z ONLY Z AND DZ1 CR Z, DZ1, AND DZ2 CCLUMNS TO ZERO DEPENDING ON INDD INDICATOR.

REFERENCES

1. Miller, R. D.; Richard, M.; and Rogers, J. T.: Feasibility of Implementing Unsteady Aerodynamics into the FLEXSTAB Computer Program System. NASA CR-132530, October 1974.

2. Miller, R. D.; Kroll, R. I.; and Clemmons, R. E.: Dynamic Loads Analysis System (DYLOFLEX) Summary. NASA CR-2846-1, 1979.

3. Hirayama, M. Y.; Kroll, R. I.; and Clemmons, R. E.: Modal Interpolation Program - L215(INTERP) Volume II - Supplemental System Design and Maintenance Document. NASA CR-2848, 1979.

4. Harrison, B. A.; and Richard, M.: A Program to Compute Three-Dimensional, Subsonic Unsteady Aerodynamic Characteristics Using the Doublet Lattice Method - L216 (DUBFLX) Volume II - Supplemental System Design and Maintenance Document. NASA CR-2850, 1979.

5. Clemmons, R. E.; and Kroll, R. I.: A Computer Program to Generate Equations of Motion Matrices - L217 (EOM) Volume II - Supplemental System Design and Maintenance Document. NASA CR-2852, 1979.

6. Anderson, L. R.; and Miller, R. D.: A Program for Calculating Load Coefficient Matrices Utilizing the Force Summation Method - L218 (LOADS) Volume II - Supplemental System Design and Maintenance Document. NASA CR-2854, 1979.

7. Hirayama, M. Y.; Clemmons, R. E.; and Miller, R. D.: Equation Modifying Program - L219 (EQMOD) Volume II - Supplemental System Design and Maintenance Document. NASA CR-2856, 1979.

8. Heidergott, K. W.: Linear Systems Analysis Program, L224 (QR). Volume II: Supplemental System Design and Maintenance Document. NASA CR-2862, 1979.

9. Graham, M. L.; Clemmons, R. E.; and Miller, R. D.: Random Harmonic Analysis Program - L221 (TEV156) Volume II - Supplemental System Design and Maintenance Document. NASA CR-2858, 1979.

10. Thornallyay, A.; Clemmons, R. E.; and Kroll, R. I.: Time History Solution Program - L225 (TEV126) Volume II - Supplemental System Design and Maintenance Document. NASA CR-2860, 1979.

11. Dusto, A. R.; Hink, G. R.; et al: A Method for Predicting the Stability Characteristics of an Elastic Airplane. NASA CR-114712-114715, Volumes 1 through 4, 1974.

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA-CR-2846-2		

4. Title and Subtitle

DYNAMIC LOADS ANALYSIS SYSTEM (DYLOFLEX) SUMMARY – VOLUME II: SUPPLEMENTAL SYSTEM DESIGN INFORMATION

5. Report Date

September 1979

6. Performing Organization Code

7. Author(s)

R. D. Miller, R. I. Kroll, and R. E. Clemmons

8. Performing Organization Report No.

D6-44455

9. Performing Organization Name and Address

Boeing Commercial Airplane Company
P.O. Box 3707
Seattle, Washington 98124

10. Work Unit No.

11. Contract or Grant No.

NAS1-13918

12. Sponsoring Agency Name and Address

National Aeronautics and Space Administration
Washington, DC 20546

13. Type of Report and Period Covered

May 1975 – May 1977

14. Sponsoring Agency Code

15. Supplementary Notes

Langley Technical Monitors: Robert C. Goetz and Boyd Perry III
Topical Report

16. Abstract

This document describes the dynamic loads analysis system DYLOFLEX. DYLOFLEX was developed to expand the aeroelastic cycle from that in the FLEXSTAB computer program system to include dynamic loads analyses involving active controls. Two aerodynamic options exist within DYLOFLEX. The analyst can formulate the problem with unsteady aerodynamics calculated using the doublet lattice method or with quasi-steady aerodynamics formulated from either FLEXSTAB or doublet lattice steady state aerodynamics with unsteady effects approximated by indicial lift growth functions. The equations of motion are formulated assuming straight and level flight and small motions. Loads are calculated using the force summation technique.

DYLOFLEX consists of nine standalone programs which can be linked with each other by magnetic files used to transmit the required data between programs.

Volume I of this document provides a brief engineering description of DYLOFLEX. Volume II contains supplemental information concerning the design and use of the program system.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Active controls Dynamic loads Quasi-steady aerodynamics Unsteady aerodynamics	Unclassified – Unlimited Subject Category 02

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	58	$5.25

*For sale by the National Technical Information Service, Springfield, Virginia 22151

NASA-Langley, 1979